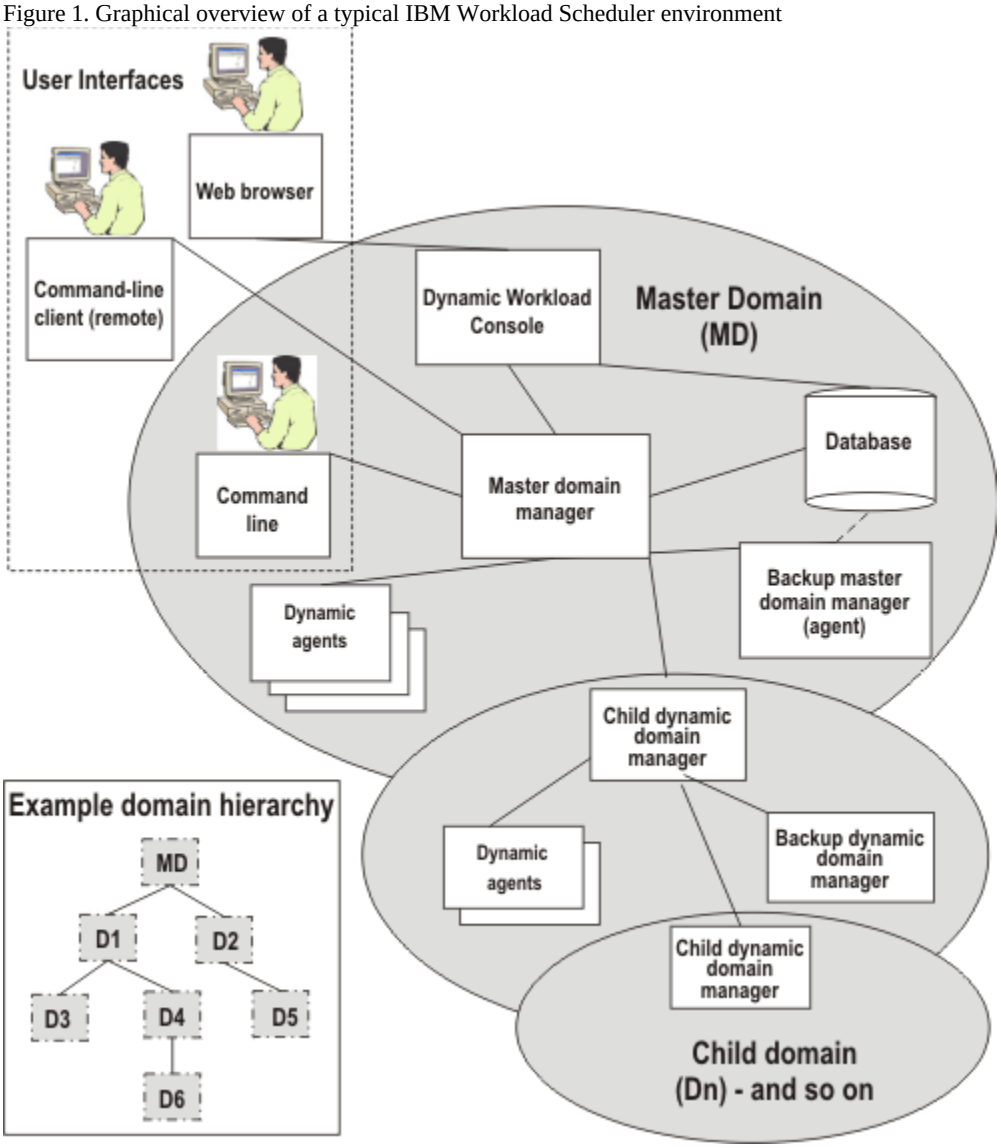
IBM® Workload Scheduler
Planning and Installation
Version 10.2.3

Part I. Planning your IBM® Workload Scheduler environment

IBM® Workload Scheduler orchestrates unattended, scheduled, and event-driven tasks for business and IT processes across on-premises and cloud environments organized in a network. A network consists of a set of linked workstations on which you perform job scheduling and processing to automate and manage your workflows. An IBM® Workload Scheduler network is composed of a master domain manager, one or more Dynamic Workload Console servers, dynamic domain managers, and dynamic agents. You might also have fault-tolerant agents, extended agents, standard agents connected to the master domain manager or to domain managers.

Figure 1: Graphical overview of a typical IBM Workload Scheduler environment *(on page 2)* gives a graphical overview of a typical IBM Workload Scheduler environment:

Figure 1. Graphical overview of a typical IBM Workload Scheduler environment



In Figure 1: Graphical overview of a typical IBM Workload Scheduler environment *(on page 2)* the master domain is shown with the main components to run your workload, and two levels of subdomain. The available user interfaces are also indicated. An example is provided of the basic domain hierarchical structure, where each domain is named "D1", "D2, and so on. All of these concepts are explained in the following section.

IBM® Workload Scheduler features the following components:

Master domain manager

The master domain manager is the highest level workstation of an IBM Workload Scheduler network. It contains or connects to the relational database that stores scheduling object definitions. It creates or updates a production plan when the plan is created or extended and then distributes the plan to the network. It performs all logging and reporting for the network. It can perform the role of event processing server for the event-driven workload automation feature.

Backup master domain manager

Define a backup master domain manager at installation to point to either the database being used by the master domain manager or to a mirror of that database. In this way the backup master domain manager has the latest data available to it at all times and can take over the role of master domain manager seamlessly, in case the master becomes unavailable.

Dynamic domain manager

Install this component if you need a multi-domain network . All domains below the master domain have dynamic domain managers to manage the workstations in their domains. Each dynamic domain manager is an agent in the domain of the next higher level. All communications to and from the dynamic agents in the domain are routed through the dynamic domain manager. To define a dynamic domain manager, install a dynamic domain manager and then perform the Configuring a dynamic domain manager *(on page 143)* procedure.

Backup dynamic domain manager

Install this component if you want a backup to your dynamic domain manager. The backup points to either the database being used by the dynamic domain manager or to a mirror of that database. If your dynamic domain manager experiences problems, you can switch to it with a simple procedure.

Agent

An agent is a workstation in the network that runs the jobs which are controlled by the IBM Workload Scheduler master domain manager. Several types of agents are available, as follows:

Dynamic agent

An agent that has the following capabilities:

Run workload dynamically

It communicates with the server the status of its resources. In this way the product is able to dynamically run your workload to the best available resources by:

- Automatically discovering scheduling environment resources.
- Automatically following resource changes
- Requesting additional resources when needed
- Matching job requirements to available resources
- Controlling and optimizing use of resources

The characteristics listed above provide high availability and load balancing potentialities to your environment and well suit virtualized environments.

When a job is submitted, either as part of a job stream in the plan or through ad hoc submission, IBM Workload Scheduler checks the job requirements, the available resources and the related characteristics and submits the job to the resource that best meets the requirements to run it.

Manage dynamic workload broker logical resource

It can remotely run, from the agent, the dynamic workload broker **resource** command on the server. To manage the **resource** command you must also install the Java™ run time.

After installing the agent, you define its type by using Configuring a dynamic agent *(on page 145)*.

In a simple configuration, dynamic agents connect directly to the master domain manager or to the dynamic domain manager. However, in more complex network topologies, if the network configuration prevents the master domain manager or the dynamic domain manager from directly communicating with the dynamic agent, for example, if the agents are behind a firewall and need to communicate through the internet, or if they need to communicate with a Network Address Translation (NAT) process, then you can configure your dynamic agents to use a local or remote gateway. In this way, communication is concentrated in a single connection, reducing the number of connections to the master domain manager or to the dynamic domain manager. For more information about the gateway parameters specified when installing a dynamic agent, see Agent installation parameters - twsinst script *(on page 84)*.

For more information about gateway configuration, see Configuring dynamic agent communications through a gateway *(on page)*.

Extended agent

Extended agents are logical definitions (hosted by a physical workstation) used to extend job processing to selected applications (SAP R/3, PeopleSoft, and z/OS®). For information about installing an extended agent, see Installing agents *(on page 79)* .

Fault-tolerant agent

A fault-tolerant agent can resolve local dependencies and launch jobs in the absence of a domain manager. It has a copy of the production control file. This allows fault-tolerant agents to continue processing even if the dynamic domain manager or the network connection is down. With a simple reconfiguration, they can serve as subordinate *domain manager*s. To define a fault-tolerant agent, install a fault-tolerant agent on your workstation and then define it as fault-tolerant in the workstation definition.

Standard agent

An agent that launches jobs only under the direction of its domain manager. It is not fault-tolerant. To define a standard agent, install a fault-tolerant agent on your workstation and then define it as a standard agent in the workstation definition.

Chapter 1. IBM Workload Scheduler interfaces

The IBM Workload Scheduler features several user interfaces from which you can manage your production environment.

You can manage your production environment from the following user interfaces:

Master domain manager command lines

The master domain manager command lines are installed automatically when you install the master domain manager. This command lines interface are run only from the workstation serving as the master domain manager. From the command lines, you can administer the master specific binaries and options. A backup master domain manager command lines also exist on the master domain manager configured as backup instance.

Dynamic Workload Console

The web-based interface for creating, modifying, monitoring, controlling, and deleting IBM Workload Scheduler objects. You can interface with the console from any system in the network where a supported web browser is installed. When you install a Dynamic Workload Console also the **z/OS® Connector** is installed, which is a component that connects IBM Z Workload Scheduler and the Dynamic Workload Console. For more information, see *IBM Z Workload Scheduler: Planning and Installation Guide*.

Integrations available on Automation Hub

Automation Hub provides an ever-growing number of integrations, software components that enable you to integrate third-party processes into the Dynamic Workload Console and enhance your automation capabilities. A great solution to automate your business workflows and manage all your processes from a single point of control. Check out the full collection at Automation Hub.

Orchestration CLI (OCLI)

Orchestration CLI is a stand-alone command-line application that you can download and install independently without requiring any other IBM® Workload Scheduler component. You can install Orchestration CLI on any workstation where you want to manage and control workflows. It is designed to replace the composer and conman commands, by providing a more modern, efficient, and versatile interface. By using Orchestration CLI, you can automate tasks efficiently, reducing manual effort and operational overhead. Orchestration CLI helps you streamline command-line interactions, enhance cross-platform compatibility, and build a more efficient workload automation process. It also simplifies maintenance, lowers costs, and minimizes IT requirements.

Orchestration CLI also provides a more modern and user-friendly interface, and is designed to be intuitive and efficient, making it easier for administrators and users to complete tasks. It combines modernity, compatibility, and enhanced functionality, when compared to the conman and composer command line.

Command-line client

A component of IBM Workload Scheduler installed only with a fault-tolerant agent that allows you to implement the following commands on the master domain manager from another workstation: The commands you can use are the following:

- Composer
- Optman
- Planman showinfo and unlock (the other planman commands must be run locally on the master domain manager)

dynamic workload broker command line

Installed and configured automatically when you install a master domain manager. It includes commands to directly submit and manage jobs for dynamic scheduling, manage job JSDL definitions and resources, and more. For more information, see Using utility commands in the dynamic environment *(on page)*

Chapter 2. Planning the environment

Typical installation scenarios for products and components.

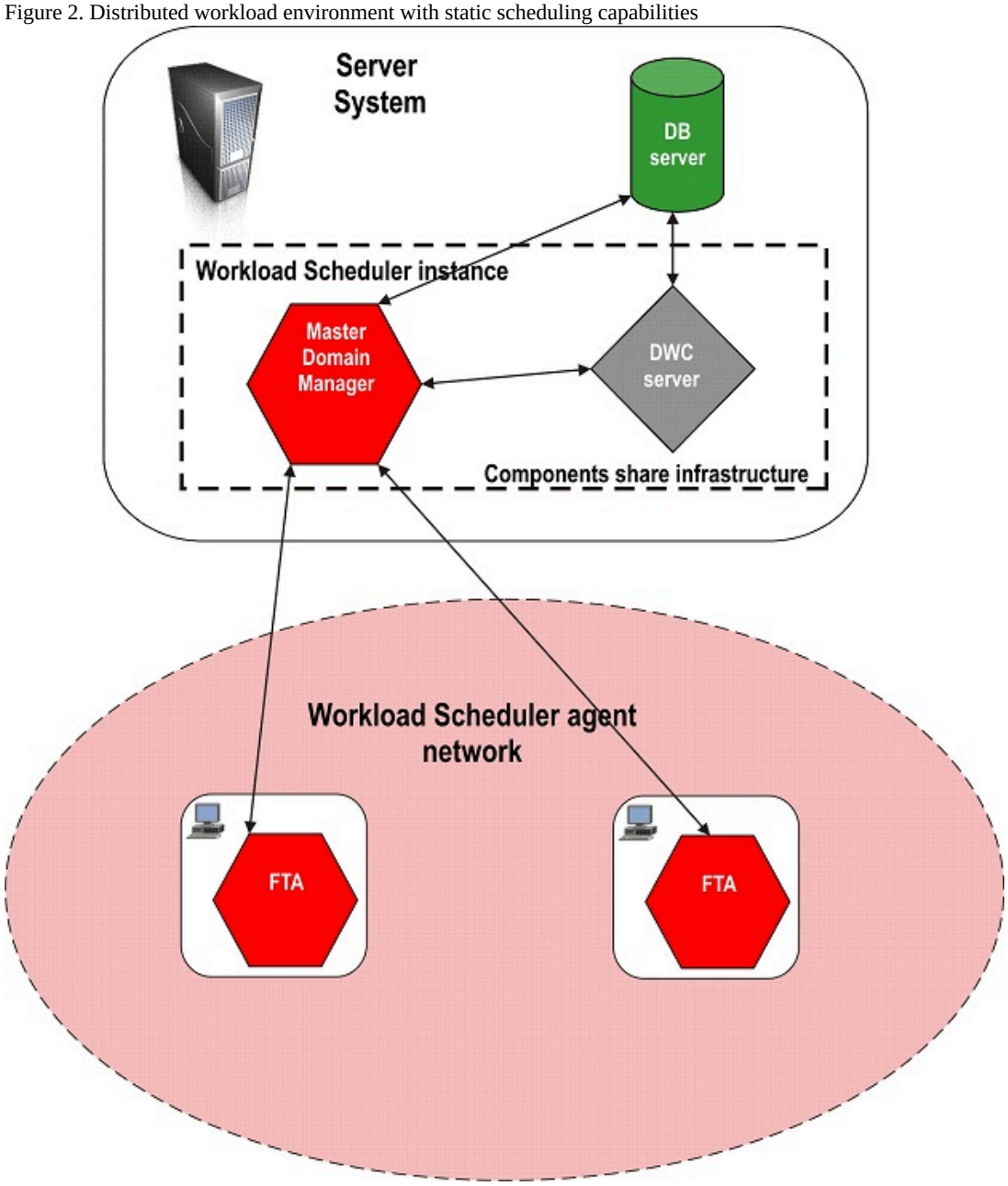
These typical scenarios for IBM Workload Automation show how to deploy specific solutions on the minimum possible system resources.

Distributed workload environment with static scheduling capabilities

Configuration to run workload statically across your distributed network.

Use this configuration to run workload statically across your distributed network. Figure 2: Distributed workload environment with static scheduling capabilities *(on page 7)* shows the system resources needed to install a fully-working IBM Workload Scheduler environment for managing your distributed workload.

Figure 2. Distributed workload environment with static scheduling capabilities



Distributed workload environment with dynamic scheduling capabilities

Use this configuration to run workload dynamically across your distributed network.

The run time environment is used to:

- Run on the agent job types with advanced options, both those supplied with the product and the additional types implemented through the custom plug-ins.
- Enable the capability to remotely run, from the agent, the dynamic workload broker resource command on the server.

For information about dynamic scheduling, how to run application job plug-ins and the dynamic workload broker resource command on the server, see *IBM Workload Scheduler: Scheduling Workload Dynamically*.

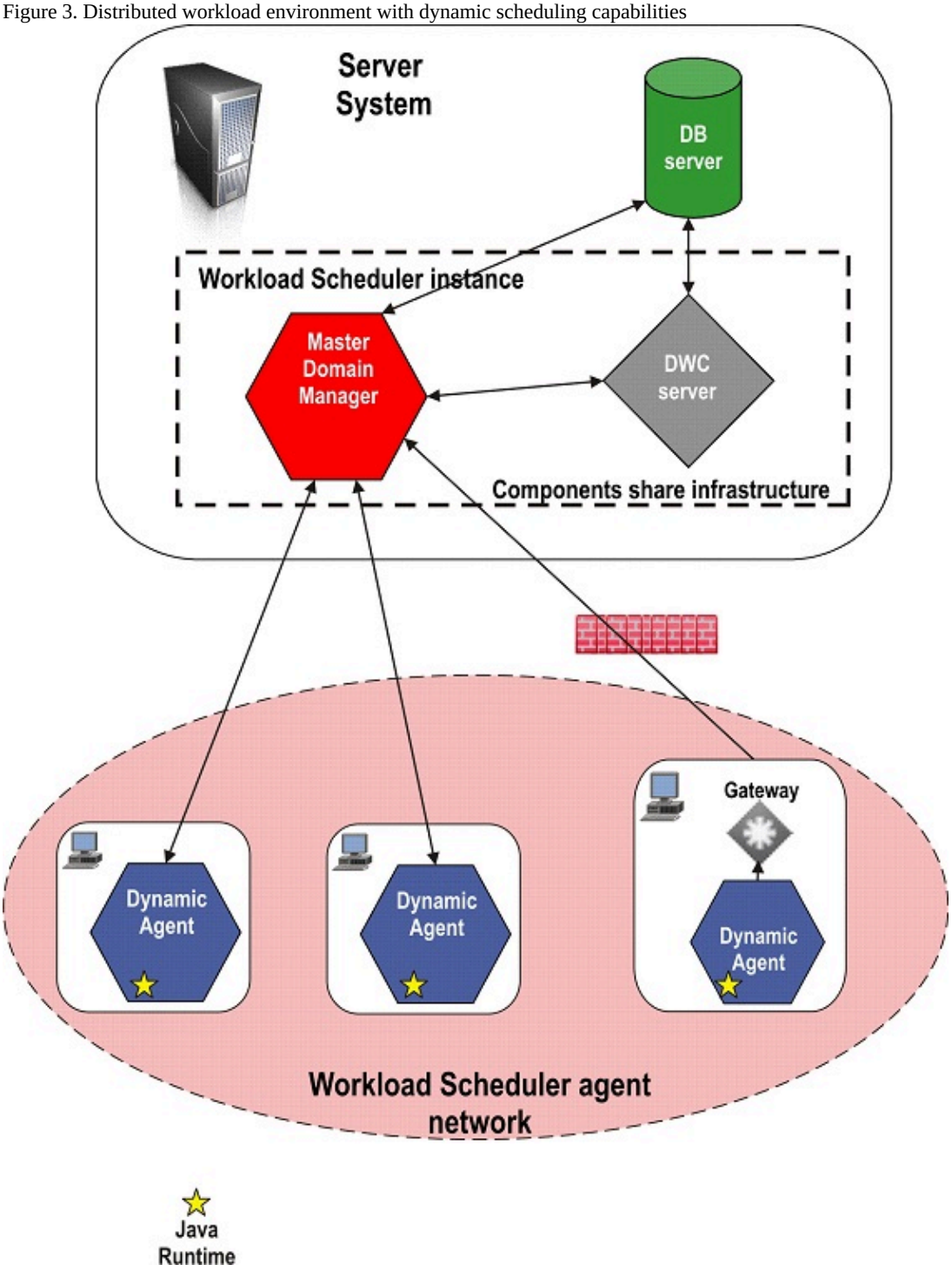
In this configuration, you can choose whether or not to add the run time environment for Java™ jobs to the agent.

Figure 3: Distributed workload environment with dynamic scheduling capabilities *(on page 9)* shows the system resources required to install a fully working IBM Workload Scheduler environment for running your distributed workload dynamically.

Note: A dynamic agent can be directly connected to its master domain manager or through a dynamic domain manager as shown in Distributed workload environment with static and dynamic scheduling capabilities *(on page 10)*. In more complex network topologies where the master domain manager or the dynamic domain manager cannot directly communicate with the dynamic agent, you can configure your dynamic agents to use a local or remote gateway. For more information about the gateway parameters specified when installing a dynamic agent, see Agent installation parameters - twsinst script *(on page 84)*. For more information about the gateway parameters specified when installing a dynamic agent, see Agent installation parameters - twsinst script *(on page 84)*.

For more information about gateway configuration, see Configuring dynamic agent communications through a gateway *(on page)* in the network communications information in the *Administration Guide*.

Figure 3. Distributed workload environment with dynamic scheduling capabilities



Dynamic scheduling supports most of the IBM Workload Scheduler features for static scheduling. The Table 1: Features partially or not supported for dynamic scheduling *(on page 9)* lists some features or properties that are partially or not supported.

Table 1. Features partially or not supported for dynamic scheduling

Feature	agent and IBM® Z Workload Scheduler agent
Event-driven workload automation.	`TivoliWorkloadSchedulerObjectMonitor` events supported.

Table 1. Features partially or not supported for dynamic scheduling (continued)

Feature	agent and IBM® Z Workload Scheduler agent
Note: For more details about the events type, see *IBM Workload Scheduler User's Guide and Reference: Appendixes - Event-driven workload automation event and action definitions*	`FileMonitor` events supported, except for IBM i systems.
	`TivoliWorkloadSchedulerApplicationMonitor` events not supported.
Utility commands (datecalc, jobinfo, and so on).	Not supported.

Distributed workload environment with static and dynamic scheduling capabilities

Use this configuration to run workload both statically and dynamically across your distributed network.

The run time environment is used to:

- Run on the agent job types with advanced options, both those supplied with the product and the additional types implemented through the custom plug-ins.
- Enable the capability to remotely run, from the agent, the dynamic workload broker resource command on the server.

For information about dynamic scheduling, how to run application job plug-ins and the dynamic workload broker resource command on the server, see *IBM Workload Scheduler: Scheduling Workload Dynamically*.

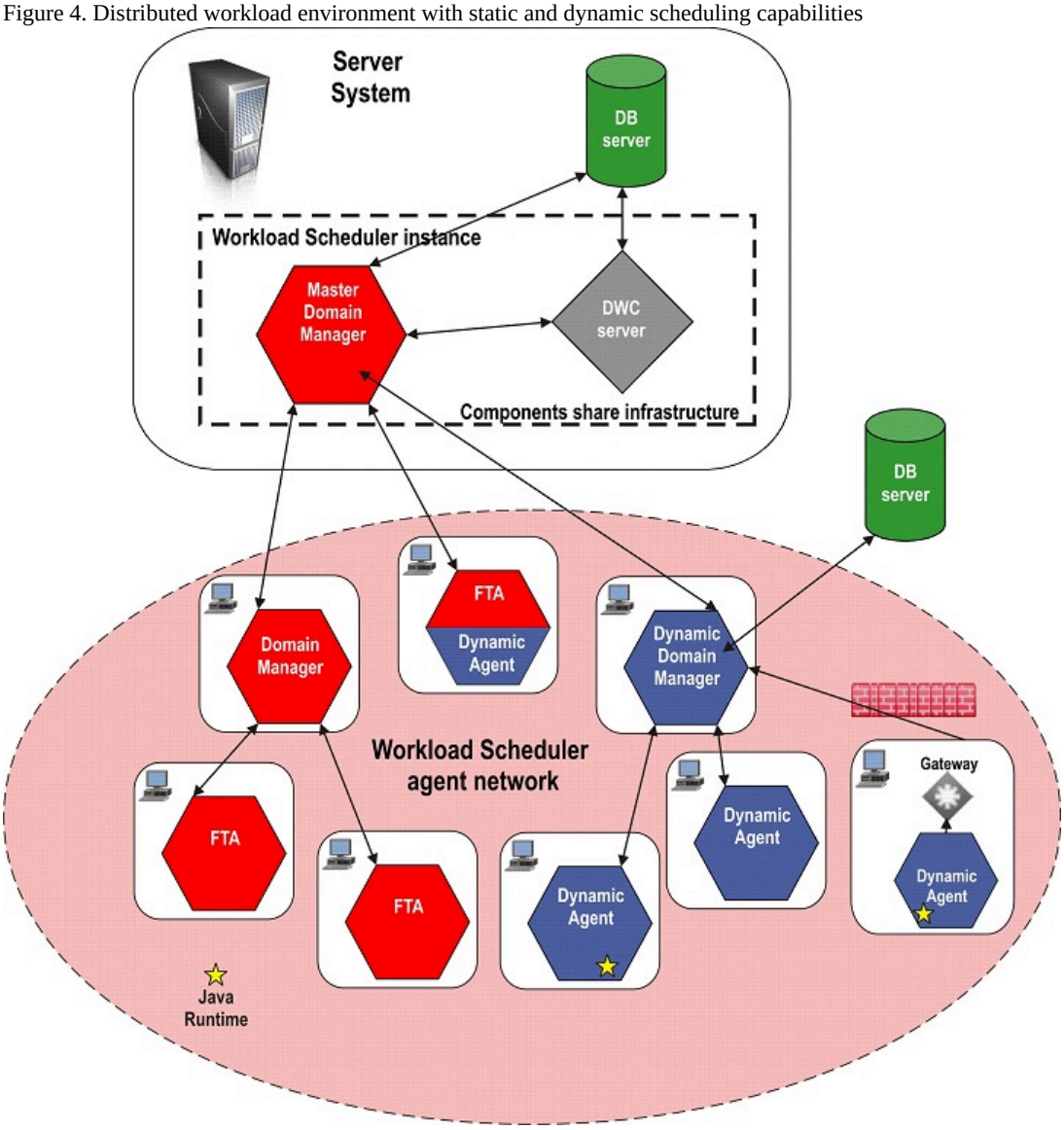
In this configuration, you can choose whether or not to add the run time environment for Java™ jobs to the agent.

Figure 4: Distributed workload environment with static and dynamic scheduling capabilities *(on page 11)* shows the system resources required to install a fully working IBM Workload Scheduler environment for running your distributed workload both statically and dynamically. IBM Workload Scheduler requires a fault-tolerant agent and a dynamic agent to be installed on every system where jobs are to scheduled statically or dynamically.

Note: A dynamic agent can be directly connected to its master domain manager or through a dynamic domain manager as shown in Figure 4: Distributed workload environment with static and dynamic scheduling capabilities *(on page 11)*. In more complex network topologies where the master domain manager or the dynamic domain manager cannot directly communicate with the dynamic agent, you can configure your dynamic agents to use a local or remote gateway. For more information about the gateway parameters specified when installing a dynamic agent, see Agent installation parameters - twsinst script *(on page 84)*.

For more information about gateway configuration, see Configuring dynamic agent communications through a gateway *(on page)* in the network communications information in the *Administration Guide*.

Figure 4. Distributed workload environment with static and dynamic scheduling capabilities



For a list of features partially or not supported in a mixed environment, see Table 1: Features partially or not supported for dynamic scheduling *(on page 9)*.

End-to-end workload environment

In an end-to-end workload environment (agent connected to the z/OS® system), you can define different types of configurations.

You can define the following types of configurations:

To run your workload statically:

Using fault-tolerant agents

Use the fault-tolerant end-to-end scheduling environment to schedule and control static workload from the mainframe to distributed systems. On the distributed system, you install fault-tolerant agents and connect them to the z/OS® server. For details, see *Scheduling End-to-end with Fault Tolerance Capabilities*.

Using IBM Z Workload Scheduler Agents (z-centric)

Use the z-centric end-to-end scheduling environment to schedule and control static workload from the mainframe to distributed systems with a low cost of ownership. On the distributed system, you install IBM Z Workload Scheduler Agents and connect them to the z/OS® controller.

For information about how to install the IBM Z Workload Scheduler Agent, see *IBM Z Workload Scheduler: Planning and Installation*. For information about how to use the IBM Z Workload Scheduler Agent, see *Scheduling End-to-end with z-centric Capabilities*.

To run your workload dynamically:

Using IBM Z Workload Scheduler Agents (z-centric) with dynamic capabilities

Use the z-centric end-to-end scheduling environment to schedule and control dynamic workload from the mainframe to distributed systems with a low cost of ownership. On the distributed system, you install IBM Z Workload Scheduler Agents , add dynamic scheduling capabilities and connect them to a dynamic domain manager that must be connected to the z/OS® controller. For information about how to:

- Install a dynamic domain manager see Installing dynamic domain components *(on page 101)*.
- Install IBM Z Workload Scheduler Agents, see *IBM Z Workload Scheduler: Planning and Installation*.
- Use IBM Z Workload Scheduler Agents, see *Scheduling End-to-end with z-centric Capabilities*.

Workload environment integrated with external systems

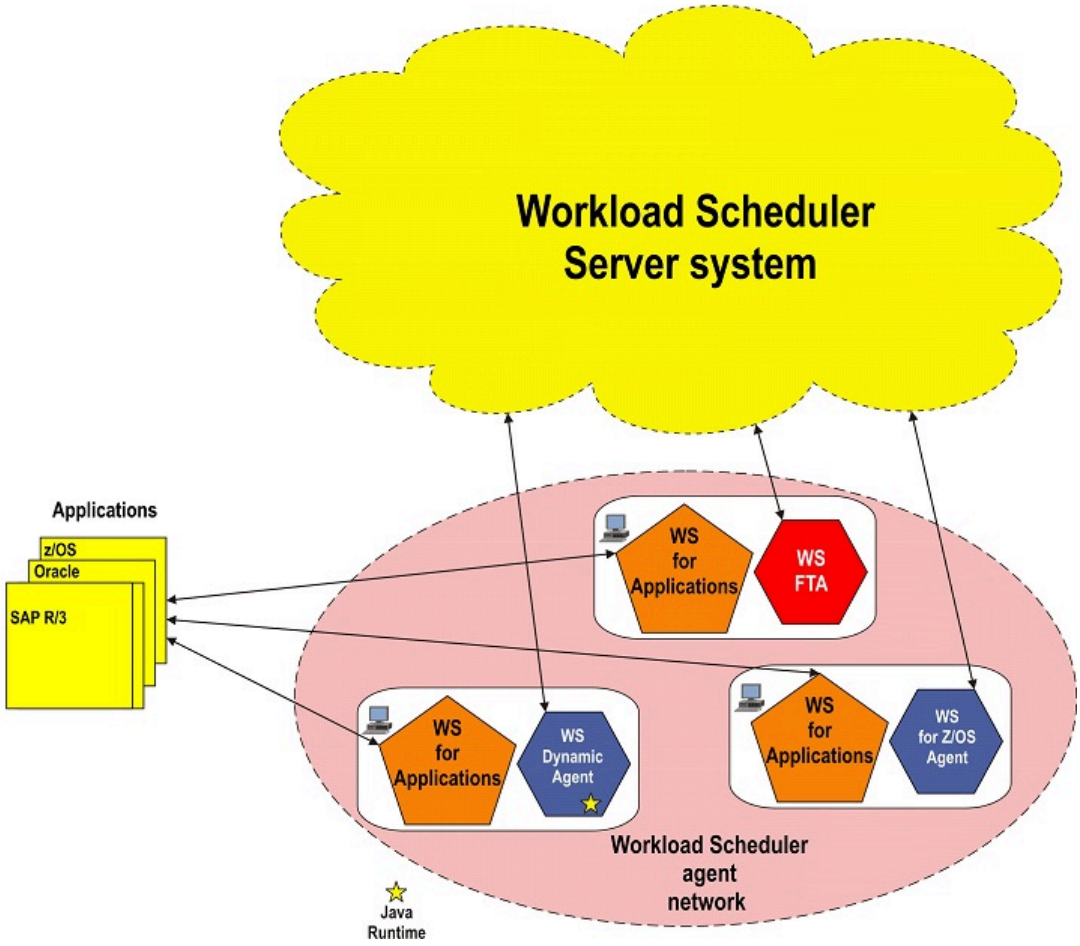
Configuration to extend IBM Workload Scheduler capabilities for scheduling on external applications.

Use this configuration to extend IBM Workload Scheduler capabilities for scheduling on external applications, such as SAP and PeopleSoft using IBM Workload Scheduler.

Figure 5: Workload environment integrated with external systems *(on page 13)* shows a sample environment including the agents needed to extend IBM Workload Scheduler scheduling capabilities on one or more external applications using IBM Workload Scheduler. You can install IBM Workload Scheduler on the master domain manager, on a fault-tolerant agents, on dynamic agents, and on IBM Z Workload Scheduler Agents.

For information about IBM Workload Scheduler, see the *IBM Workload Scheduler: User's Guide* documentation.

Figure 5. Workload environment integrated with external systems



Note: Installing IBM Workload Scheduler on an agent (master domain manager, domain manager, fault-tolerant agent, standard agent, dynamic agent, IBM Z Workload Scheduler Agent) is the correct deployment scenario in an end-to-end environment.

Distributed-driven workload environment for z/OS®

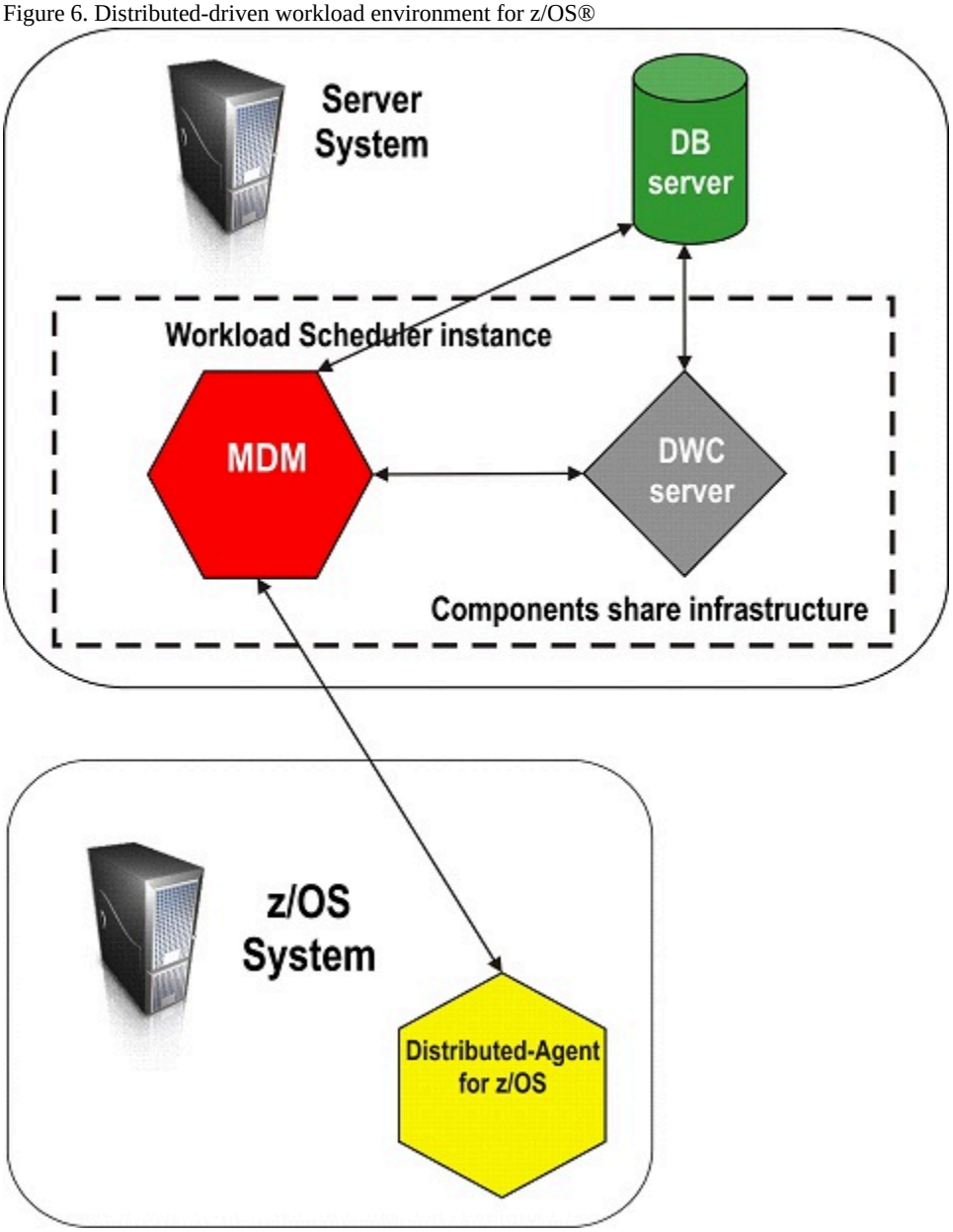
Configuration used when submitting from the IBM Workload Scheduler.

Use this configuration to submit from the IBM Workload Scheduler (using the dynamic workload broker component installed with the master domain manager or the dynamic domain manager) workload to be processed by JES2, without having to define the workload on the z/OS® system.

Figure 5: Workload environment integrated with external systems *(on page 13)* shows the minimum system resources needed to install a distributed-driven environment, where the IBM Workload Scheduler distributed-Agent for z/OS® represents a lightweight end-to-end scheduling solution where you define and manage on the distributed side the workload that is to be processed by JES2.

For information about IBM Workload Scheduler distributed-Agent for z/OS®, see the *IBM Workload Scheduler: Scheduling with the Agent for z/OS* documentation.

Figure 6. Distributed-driven workload environment for z/OS®



Dockerized environment

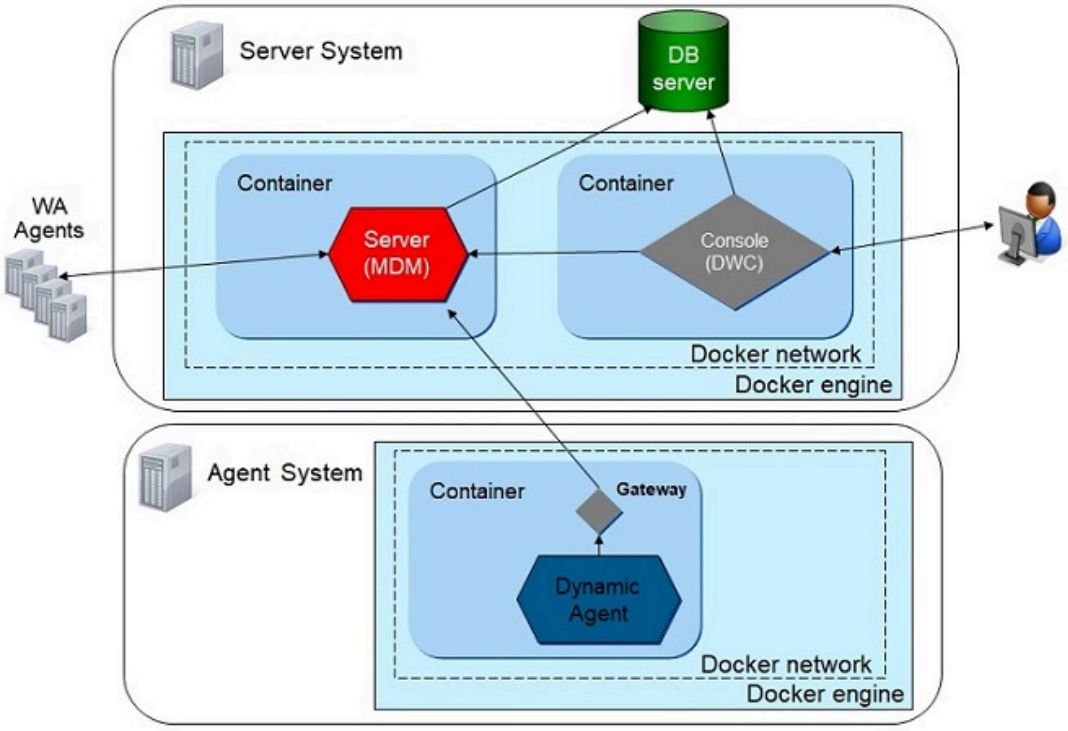
Use this configuration to implement a Dockerized environment.

Use this configuration to benefit of the IBM Workload Automation on a dockerized environment. Three containers are delivered and they can be deployed on the same engine or on different ones.

In the *Figure 1*, server and console components have been deployed on the same Docker engine and the dynamic agent component on a separated engine.

The database is always external to the Docker engine and a connection is established with server and console.

Figure 7. Dockerized environment configuration



Chapter 3. Planning domains

A IBM Workload Scheduler network contains at least one master domain manager that acts as a management hub for the product. Additional domains can be used to divide a widely-distributed network into locally-managed groups of workstations.

In a single domain configuration, the master domain manager maintains communications with all of the workstations in the network.

In a multiple domain configuration, the master domain manager communicates with the workstations in its domain and all immediately subordinate domain managers. The subordinate domain managers communicate with the workstations in their domains and their immediately subordinate domain managers, and so on. Domain managers report all of the activities of the domain to the master. Using multiple domains reduces network traffic and the load on the master by reducing the number of direct communications between the master domain manager and workstations. Multiple domains also provide fault-tolerance by limiting the outage caused by losing a domain manager in a single domain. To limit the effects further, you can designate backup domain managers to take over if domain managers fail.

When you define a new domain, you must identify the parent domain and the domain manager. The parent domain is the domain directly above the new domain in the domain hierarchy. All communications to and from a domain are routed through the parent domain manager.

Localized processing in your domain

Localized processing is separating your scheduling needs based on a common set of characteristics, such as geographical locations, business functions, and application groupings.

Group related processing can limit the amount of interdependency information that needs to be communicated between domains. The benefits of localized domains are:

Decreased network traffic

Keeping processing localized to domains eliminates the need for frequent inter-domain communication.

Tighter security and simplified administration

Security and administration can be defined at and limited to the domain level. Instead of network-wide or workstation-specific administration, you can have domain administration.

Optimized network and workstation fault-tolerance

In a multiple domain network, you can define backups for each domain manager so that problems in one domain do not disrupt operations in other domains.

Considerations in planning domains

There are a number of considerations that are to be taken into account when planning domains.

In planning your IBM Workload Scheduler network, consider the following:

Number of workstations, applications, and jobs

Consider the number of workstations that comprise the network and the number of applications and jobs that the network runs. If you have a small number of workstations, or a small number of applications to control, you do not need multiple domains.

Number of geographic locations

Consider the number of geographic locations covered by your network and the reliability and efficiency of communication between the locations. Multiple geographic locations is one of the primary reasons for choosing a multiple domain architecture. One domain for each geographical location is a common configuration. A single domain architecture relies on the network maintaining continuous processing.

Time zones

When your network is spread across multiple geographic locations in different time zones, decide whether to activate the time zone feature. See Time zone considerations *(on page 17)*.

Centralized or decentralized management

You can manage single or multiple domain networks from a single master domain manager. If you want to manage multiple locations separately, you can consider the installation of a separate IBM Workload Scheduler network at each location. Some decentralized management is possible in a stand-alone IBM Workload Scheduler network by mounting or sharing file systems.

Types of applications

Consider the types of applications that are run by IBM Workload Scheduler. If you have multiple applications that are distinctly separate from each other, you might choose to put them in separate domains.

Windows™ network

When you have a Windows™ network, you might want your IBM Workload Scheduler domains to mirror your Windows™ domains.

System performance and other criteria

You can define multiple domains to localize systems based on performance or operating system type.

Amount of network traffic

If your network traffic is manageable, having multiple domains is less important.

Dependencies between jobs

Consider if you need to plan for job dependencies that cross system boundaries, geographical boundaries, or application boundaries. For example, does the start of Job1 on workstation1 depend on the completion of Job2 running on workstation2. The degree of interdependence between jobs is an important consideration when planning your network. If you use multiple domains, try to keep interdependent objects in the same domain to decrease network traffic and improve the use of the domain architecture. See *User's Guide and Reference*.

Level of fault-tolerance required

A disadvantage of the single domain configuration is the reliance on a single domain manager. In a multi-domain network, the loss of a single domain manager affects only the agents in its domain.

Firewalls

When your network contains firewalls, plan the structure of your domains around the firewalls. See *Administration Guide*.

Workstation classes

Workstations are organized into domains to make your network management easier and more efficient. However, the domain name is not one of the selection criteria when choosing where to run a job or job stream.

If you want to group workstations together because they have similar job scheduling characteristics, use a workstation class. Any number of workstations can be grouped in a class, and a workstation can be in many classes. Jobs and job streams can be assigned to run on a specific workstation class.

For example, you could set up workstation classes to group workstations according to:

- Your internal departmental structure, so that you could define a job that would be run on all the workstations in a department
- The software installed on them, so that you could define a job that would be run on all the workstations that had a particular application installed
- The role of the user, so that you could define a job that would be run on all the workstations belonging to, for example, managers

In this example, an individual workstation could be in one workstation class for its department, another for its user, and several others for the software installed on it.

Time zone considerations

Time zone support is an optional feature that is enabled by default.

It allows you to manage workloads at a global level. Time zone implementation also enables easy scheduling across multiple time zones.

For a description of how the time zone implementation works, see How IBM Workload Scheduler manages time zones *(on page)*.

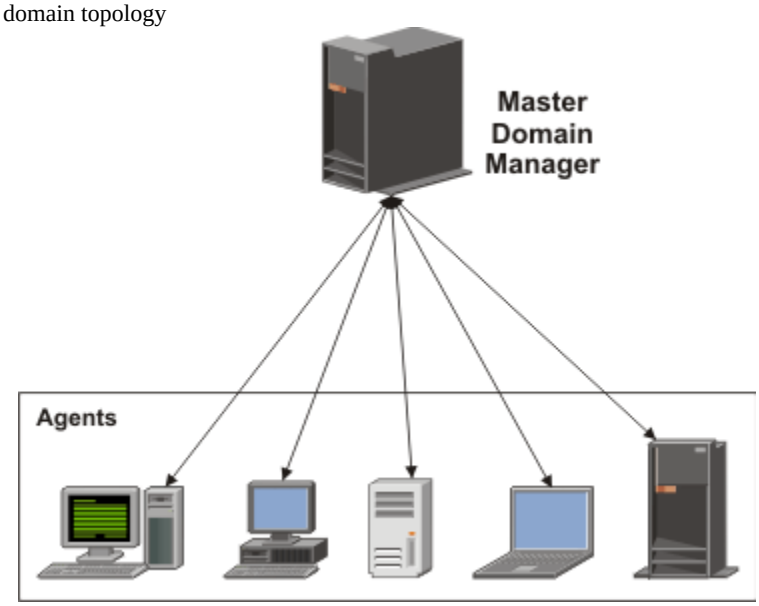
For information about how to set the time zone implementation, see Enabling the time zone feature *(on page)*.

Single domain network

A single domain network consists of a master domain manager and any number of agents.

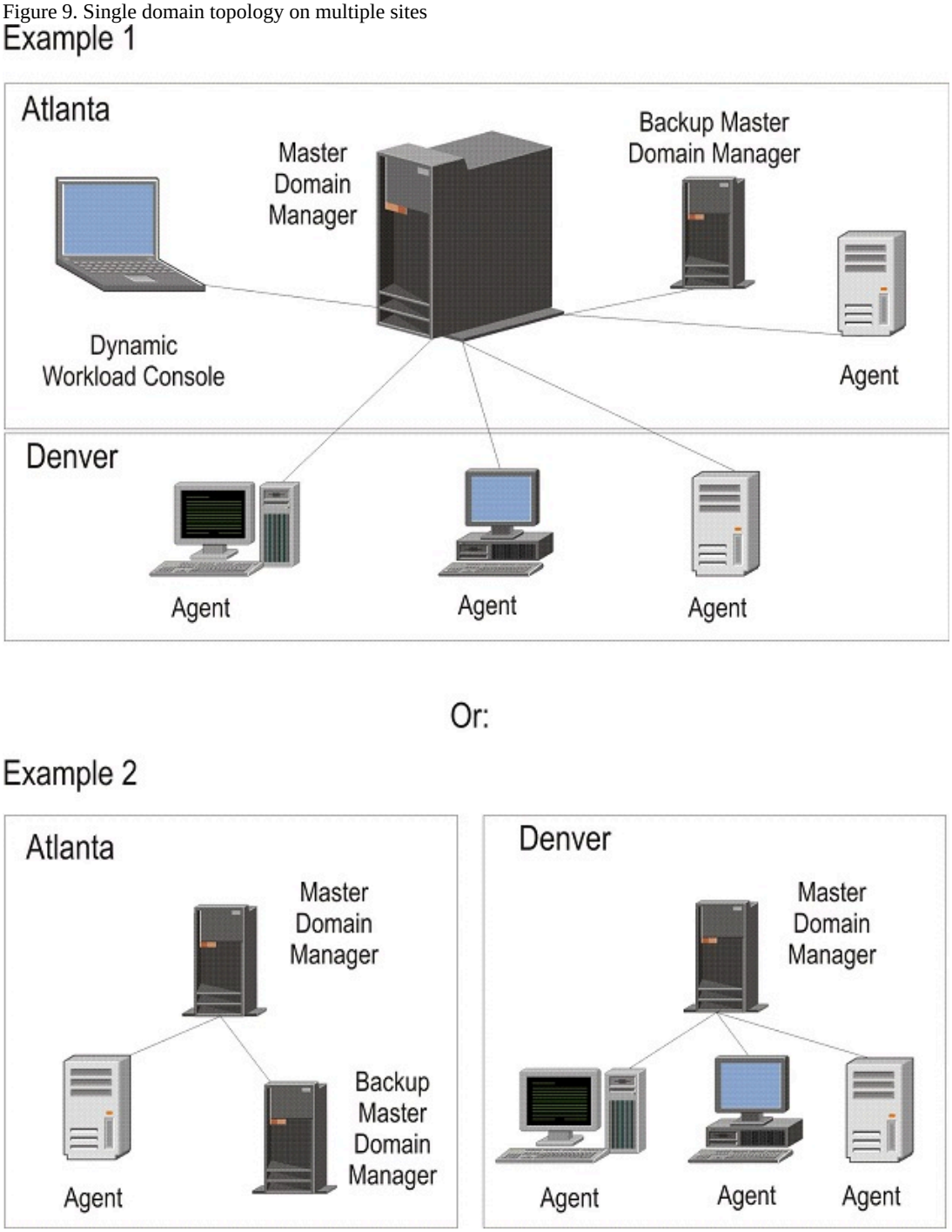
Figure 8: Single domain topology *(on page 18)* shows an example of a single domain network. A single domain network is well-suited to companies that have few locations and business functions. All communication in the network is routed through the master domain manager. With a single location, you are concerned only with the reliability of your local network and the amount of traffic it can handle.

Figure 8. Single domain topology



Single domain networks can be combined with other networks, single or multiple domain, to meet multiple site requirements. IBM Workload Scheduler supports internetwork dependencies between jobs running on different networks.

Figure 9. Single domain topology on multiple sites



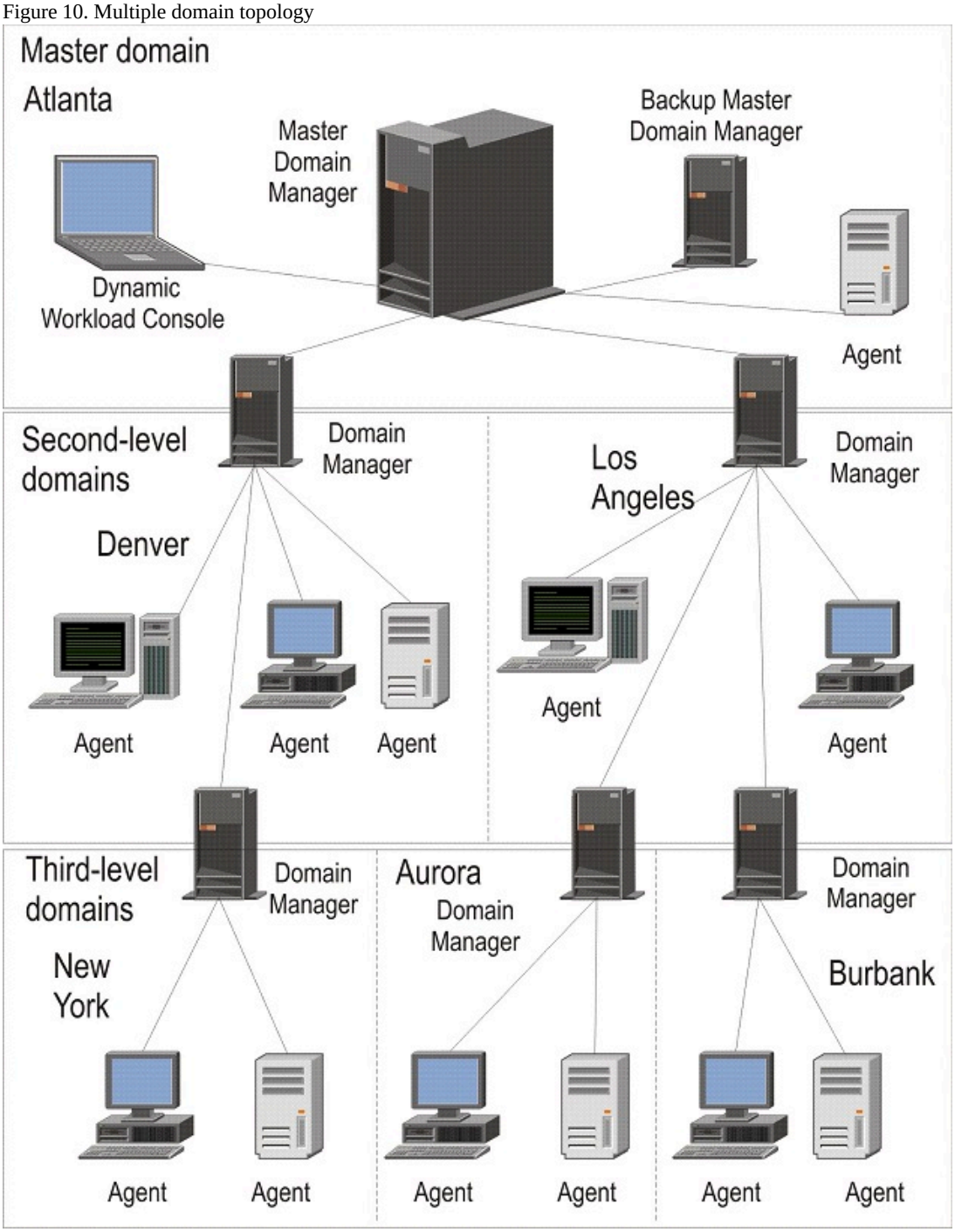
Example 1 shows a single domain network. The master domain manager is located in Atlanta, along with several agents. There are also agents located in Denver. The agents in Denver depend on the master domain manager in Atlanta to resolve all interagent dependencies, even though the dependencies might be on jobs that run in Denver. An alternative would be to create separate single domain networks in Atlanta and Denver, as shown in example 2.

Multiple domain network

Multiple domain networks are especially suited to companies that span multiple locations, departments, or business functions.

A multiple domain network consists of a master domain manager, any number of lower tier domain managers, and any number of agents in each domain. Agents communicate only with their domain managers, and domain managers communicate with their parent domain managers. The hierarchy of domains can go down to any number of levels.

Figure 10. Multiple domain topology



As Figure 10: Multiple domain topology *(on page 20)* illustrates, the master domain manager is located in Atlanta. The master domain manager contains the database files used to document the scheduling objects, and distributes the Symphony file to its agents and the domain managers in Denver and Los Angeles. The Denver and Los Angeles domain managers then distribute the Symphony file to their agents and subordinate domain managers in New York, Aurora, and Burbank. The master domain manager in Atlanta is responsible for broadcasting inter-domain information throughout the network.

All communication to and from the New York domain manager is routed through its parent domain manager in Denver. If there are schedules or jobs in the New York domain that are dependent on schedules or jobs in the Aurora domain, those dependencies are resolved by the Denver domain manager. Most inter-agent dependencies are handled locally by the lower tier domain managers, greatly reducing traffic on the network.

Chapter 4. Installation considerations

Some considerations that need to be taken into account before installation.

Before you begin the installation using the installation wizard, consider the following items that might apply to your specific environment.

Installing on Windows™ operating systems

If you are installing on Windows™, consider the following items.

- If you are using Windows™ Terminal Services, set the install user with the command: `change user /install`
- If TWS_USER is a domain user, Microsoft™ Computer Browser Service must be active.
- If TWS_USER is a domain user, the user performing the installation must be a domain administrator.

Remote installation

You cannot install IBM Workload Scheduler on a Windows™ workstation from a remote Samba-mounted file system.

Installing for end-to-end scheduling

If you are installing IBM Workload Scheduler on a workstation used as a distributed agent (that is either a standard agent, fault-tolerant agent, or domain manager) for end-to-end scheduling, specify OPCMASTER as the name of the master domain manager during the installation process. For further information about installing for end-to-end scheduling, see *Scheduling End-to-end with Fault Tolerance Capabilities*.

Create symbolic links

UNIX™ and Linux™. The installation wizard installs all executable files in its own `.bin` directory. Before running any IBM Workload Scheduler commands, you run a script that sets the command-line environment to access these files. To avoid having to set the environment each time you want to run any of the commands from within a script, you can select an installation option to create symbolic links to those commands or utilities most frequently used from within scripts. Table 2: Symbolic link options *(on page 22)* shows the binary paths and the symbolic links.

Table 2. Symbolic link options

TWS binary path	Symbolic link
TWS_home/bin/at	usr/bin/mat
TWS_home/bin/batch	usr/bin/mbatch
TWS_home/bin/datecalc	usr/bin/datecalc
TWS_home/bin/jobstdl	usr/bin/jobstdl
TWS_home/bin/maestro	usr/bin/maestro
TWS_home/bin/mdemon	usr/bin/mdemon
TWS_home/bin/morestdl	usr/bin/morestdl
TWS_home/bin/muser	usr/bin/muser
TWS_home/bin/parms	usr/bin/parms

Installation paths

IBM Workload Automation is the name of a family of products and components, which includes the following:

- IBM Workload Scheduler
- IBM® Z Workload Scheduler

- IBM Workload Scheduler for Applications
- Dynamic Workload Console

Many IBM Workload Scheduler components are installed in what is called an *IBM Workload Automation instance*.

This section describes the installation paths of the IBM Workload Scheduler components:

***TWA_home* installation path**

Many of the components are installed in an IBM Workload Automation instance. Although this is a notional structure it is represented on the computer where you install IBM Workload Automation components by a common directory referred to in the documentation as *TWA_home*. The path of this directory is determined when you install an IBM Workload Scheduler component for the first time on a computer. You have the opportunity to choose the path when you make that first-time installation, but if you accept the default path, it is as follows:

On UNIX™ operating systems

`/opt/wa/server_<wauser><n>`

On Windows™ operating systems

`%Program Files%\wa\server<n>`

where <*n*> is an integer value ranging from 0 for the first instance installed, 1 for the second, and so on.

This path is called, in the publications, *TWA_home*. For details about the directories created outside of *TWA_home*, see Directories created outside of TWA_home at installation time *(on page 27)* .

***TWA_DATA_DIR* and *DWC_DATA_dir* configuration directories**

To simplify administration, configuration, and backup and recovery on UNIX systems, a new default behavior has been implemented with regard to the storage of product data and data generated by IBM® Workload Scheduler, such as logs and configuration information. These files are now stored by default in the `<data_dir>` directory, which you can optionally customize at installation time.

By default, this directory is *TWA_home*`/TWSDATA` for the server and agent components, and *DWC_home*`/DWC_DATA` for the Dynamic Workload Console. The product binaries are stored instead, in the installation directory.

You can optionally customize the `<data_dir>` directory at installation time by setting the **--data_dir** argument when you install using the command-line installation. If you want to maintain the previous behavior, you can set the **--data_dir** argument to the IBM® Workload Scheduler installation directory.

If you deploy the product components using Docker containers, the `<data_dir>` is set to the default directory name and location, and it cannot be modified.

To retrieve the *TWA_DATA_DIR* and *DWC_DATA_dir* location in case you have modified the default path, check the values for the TWS_datadir and DWC_datadir properties stored in the `twainstance<instance_number>.TWA.properties` file. The file is located in `/etc/TWA`.

Alternatively, you can also proceed as follows:

1. Browse to `<TWA_home>/TWS` path.
2. Source the . ./tws_env.sh shell script.
3. Type echo $UNISONWORK. As a result, the path to the *TWA_DATA_DIR* is returned.

IBM Workload Scheduler installation directory

You can install more than one IBM Workload Scheduler component (master domain manager, backup master domain manager, domain manager, or backup domain manager) on a system, but each is installed in a separate instance of IBM Workload Automation, as described above.

The installation directory of IBM Workload Scheduler is:

`<TWA_home>/TWS`

***DWC_home* installation directory**

The Dynamic Workload Console can be installed in the path of your choice, but the default installation directory is as follows:

On Windows™ operating systems

`%ProgramFiles%\wa\DWC`

On UNIX™ operating systems

`/opt/wa/DWC`

On z/OS operating system

`/opt/wa/DWC`

IBM Workload Scheduler agent installation directory

The agent also uses the same default path structure, but has its own separate installation directory:

`<TWA_home>/TWS/ITA/cpa`

Note: The agent also installs some files outside this path. If you have to share, map, or copy the agent files (for example when configuring support for clustering) share, map, or copy these files, as well:

On UNIX™ operating systems

```
/etc/teb/teb_tws_cpa_agent_<tws_user>.ini
/opt/IBM/CAP/EMICPA_default.xml
  /etc/init.d/tebctl-tws_cpa_agent_<tws_user>
   (on Linux)
   /etc/rc.d/init.d/tebctl-tws_cpa_agent_<tws_user>
   (on AIX)
```

On Windows™ operating systems

```
%windir%\teb\teb_tws_cpa_agent_<tws_user>.ini
%ALLUSERSPROFILE%\IBM\CAP\EMICPA_default.xml
```

The agent uses the following configuration files which you might need to modify:

JobManager.ini

This file contains the parameters that tell the agent how to run jobs. You should only change the parameters if advised to do so in the IBM Workload Scheduler documentation or requested to do so by IBM Software Support. Its path is:

On UNIX™ operating systems

TWA_DATA_DIR`/ITA/cpa/config/JobManager.ini`

On Windows™ operating systems

TWA_home`\TWS\ITA\cpa\config\JobManager.ini`

JobManagerGW.ini

When a dynamic agent is installed and **-gateway** `local|remote` is specified, then this file contains the same parameters as the `JobManager.ini` file except for the following differences:

- The **ResourceAdvisorUrl** parameter points to the dynamic workload broker, and not the master domain manager.

The `JobManagerGW.ini` file is installed in the following location:

On UNIX™ operating systems

TWA_DATA_DIR`/ITA/cpa/config/JobManagerGW.ini`

On Windows™ operating systems

TWA_home\TWS\ITA\cpa\config\JobManagerGW.ini

ita.ini

This file contains parameters which determine how the agent behaves. Changing these parameters may compromise the agent functionality and require it to be reinstalled. You should only change the parameters if advised to do so in the IBM Workload Scheduler documentation or requested to do so by IBM Software Support. Its path is:

On UNIX™ operating systems

TWA_DATA_DIR/ITA/cpa/ita/ita.ini

On Windows™ operating systems

*TWA_home*TWS\ITA\cpa\config\ita.ini

Installation path for files giving the dynamic scheduling capability

The files that give the dynamic scheduling capability are installed in the following path:

<TWA_home>/TDWB

The command line client installation path

The command line client is installed outside all *IBM Workload Automation instances*. Its default path is:

TWA_home/TWS/CLI

However, the information above supplies only the **default** paths. To determine the actual paths of products and components installed in IBM Workload Automation instances, see Finding out what has been installed in which IBM Workload Automation instances *(on page 25)*

Finding out what has been installed in which IBM Workload Automation instances

If you are not the installer of IBM Workload Scheduler and its components, you might not know what components have been installed, and in which instances of IBM Workload Automation. Follow this procedure to find out:

1. Access the following directory:

   UNIX™ and Linux™ operating systems

   /etc/TWA

   Windows™ operating systems

   %windir%\TWA

2. List the contents of the directory. Each IBM Workload Automation instance is represented by a file called: twainstance<*instance_number*>.TWA.properties. These files are deleted when all the products or components in an instance are uninstalled, so the number of files present indicates the number of valid instances currently in use.
3. Open a file in a text viewer.

Attention: Do not edit the contents of this file, unless directed to do so by IBM Software Support. Doing so might invalidate your IBM Workload Scheduler environment.

The contents are similar to this on a master domain manager :

```
#TWAInstance registry
#Mon Feb 26 09:28:08 EST 2024
TWA_path=/opt/wa/server_twsuser
TWA_componentList=TWS
TWS_version=10.2.3
TWS_counter=1
TWS_instance_type=MDM
TWS_basePath=/opt/wa/server_twsuser/TWS
TWS_user_name=twsuser
TWS_wlpdir=/opt/wa/wlpEngine/wlp
```

```
TWS_datadir=/opt/wa/server_twsuser/TWSDATA
TWS_jdbcdir=/opt/wa/server_twsuser/TWS/jdbcdrivers/db2
```

The contents are similar to this on the Dynamic Workload Console:

```
#TWAInstance registry
Mon Feb 26 09:28:08 EST 2024
TWA_path=/opt/wa/DWC
TWA_componentList=DWC
DWC_version=10.2.3
DWC_counter=1
DWC_instance_type=DWC
DWC_basePath=/opt/wa/DWC
DWC_user_name=dwcadmin
DWC_wlpdir=/opt/wa/wlpDWC/wlp
DWC_datadir=/opt/wa/DWC/DWC_DATA
DWC_jdbcdir=/opt/wa/DWC/jdbcdrivers/db2
```

The important keys to interpret in this file are:

TWA_path

This is the base path, to which the installation added one or more of the following directories, depending on what was installed:

`TWS`

Where the IBM Workload Scheduler component is installed

`DWC`

Where the Dynamic Workload Console is installed

`ssm`

Where the Netcool® SSM monitoring agent is installed (used in event management)

TWA_componentList

Lists the components installed in the instance of IBM Workload Automation.

TWS_counter

Indicates if an IBM Workload Scheduler component is installed in this instance of IBM Workload Automation (when the value=1).

TWS_instance_type

Indicates which component of IBM Workload Scheduler is installed in this instance:

MDM

Master domain manager

BKM

Backup master domain manager

DDM

dynamic domain manager

FTA

Fault-tolerant agent or domain manager

TWS_user_name

The ID of the *<<TWS_user>>* of the IBM Workload Scheduler component.

TWS_wlpdir

The installation directory of the WebSphere Application Server Liberty Base instance used by IBM Workload Scheduler.

TWS_datadir

The directory containing product data and data generated by IBM Workload Scheduler, such as logs and configuration information.

DWC_counter

Indicates if an instance of Dynamic Workload Console is installed in this instance of IBM Workload Automation (when the value=1)

DWC_user_name

The ID of the Dynamic Workload Console user.

DWC_wlpdir

The installation directory of the WebSphere Application Server Liberty Base instance used by Dynamic Workload Console.

DWC_datadir

The directory containing product data and data generated by Dynamic Workload Console, such as logs and configuration information.

Directories created outside of *TWA_home* at installation time

The following list shows the directories that are created outside of *TWA_home* when you install IBM Workload Scheduler.

Windows operating systems

```
%WINDIR%\TWA

 %WINDIR%\system32\TWSRegistry.dat (32 bits)
 %WINDIR%\sysWOW64\TWSRegistry.dat (32 bits on 64 bits)
 %WINDIR%\TWSRegistry.dat (64 bits on 64 bits)
 %WINDIR%teb
 %WINDiR%\cit
 %ProgramFiles%\tivoli\cit (or the path specified by %WINDiR%\cit\cit.ini)
```

UNIX

```
/etc/TWA
 /etc/TWS
 /etc/teb
 /etc/cit
 /etc/init.d/tebclt-tws_cpa_agent_instance_name
 /usr/Tivoli/TWS
 /usr/ibm/tivoli/common/CIT/logs
 /opt/tivoli/cit (or the path specified by /etc/tivoli/cit/cit.ini)
```

IBM Workload Scheduler also installs some files in the following existing folders:

```
/etc/systemd
/etc/rc.d
```

Note: If you want to check which files are stored in the **etc** directories, you can launch the following command: `find /etc -name "*<tws_user>*"`

Windows™ services

When installing on the Windows™ operating system the Windows™ Service Control Manager registers services.

An installation on Windows™ operating systems registers the following services on the Windows™ Service Control Manager:

- IBM Workload Scheduler (for *TWS_user*)
- Netman (for *TWS_user*)
- Token Service (for *TWS_user*)
- IBM Workload Scheduler SSM Agent (for *TWS_user*)
- IBM Common Platform Agent: tws_cpa_agent_ (for *TWS_user*)

Note: An existing service that has the same name as the new service will be overwritten during installation.

The Service Control Manager maintains its own user password database. If the *TWS_user* password is changed after installation, you must use the Services applet in the Control Panel to assign the new password for the Token Service and IBM Workload Scheduler (for *TWS_user*). For more information, see Changing key IBM Workload Scheduler passwords *(on page)*.

Part II. Installing IBM® Workload Scheduler

Available installation methods

This section provides the information required before you install the product. The available installation methods are listed, together with some considerations:

Advantages of the command-line installation

The command-line installation is a very simple procedure, which supports installing all components (master domain manager, backup domain manager, dynamic domain manager, backup dynamic domain manager, Dynamic Workload Console, and agents) using dedicated commands. You can choose to maintain the default values already defined in the properties file, specify all or part of the parameters in the command line when typing the command, or edit all or part of the parameters stored in the properties file. To proceed with the command-line installation, skip to Installing from the command-line interface *(on page 31)*.

Advantages of the Docker deployment

The Docker installation is comprised of a set of pre-installed images for the master domain manager, the Dynamic Workload Console, and the DB2 database. All you have to do is launch the Docker installation commands.

Docker is a state-of-the-art technology which creates, deploys, and runs applications by using containers. Packages are provided containing an application with all of the components it requires, such as libraries, specific configurations, and other dependencies, and deploy it in no time on any other Linux or Windows workstation, regardless of any different settings between the source and the target workstation.

Docker adoption ensures standardization of your workload scheduling environment and provides an easy method to replicate environments quickly in development, build, test, and production environments, speeding up the time it takes to get from build to production significantly. Install your environment using Docker to improve scalability, portability, and efficiency.

To proceed with the Docker installation, skip to Deploying containers with Docker *(on page 121)*.

Advantages of the Red Hat OpenShift deployment

The IBM Workload Automation product components can be deployed onto Red Hat OpenShift, V4.*x*. You can deploy IBM Workload Automation components using IBM® certified containers on a Kubernetes-based container application platform useful to orchestrate containerized applications. You can then manage the IBM Workload Automation containers from the OpenShift dashboard or from the command line interface.

The IBM Workload Automation agent container can be deployed onto OpenShift, V3.*x*, a Kubernetes-based container application platform useful to orchestrate containerized applications. By using OpenShift, you can deploy the IBM Workload Automation agent container with a *template.yml* file to quickly configure and run it as Docker container application in a Kubernetes cluster. You can then manage the IBM Workload Automation agent container from the OpenShift dashboard or from the command line interface.

With OpenShift, you can implement distributed, advanced and scalable services based on the Docker container technology and orchestrated by Kubernetes. For more information, see Deploying IBM Workload Automation components on Red Hat OpenShift *(on page 123)*.

Advantages of deploying on Amazon EKS

To respond to the growing request to make automation opportunities more accessible, IBM® Workload Scheduler is now offered on the Amazon Web Services cloud. Within just a few minutes, you can access the product Helm chart and container images and easily launch an instance to deploy an IBM® Workload Scheduler server, console, and agents with full on-premises capabilities on AWS. IBM® Workload Scheduler on AWS improves flexibility and scalability of your automation environment. It helps in lowering costs and eliminating complexity, while reducing the operational overhead and the burden involved in managing your own infrastructure, so you can invest your time and resources in growing your business. Also, IBM® Workload Scheduler on AWS delivers faster access to managed services solutions, for a full product lifecycle management.

For more information see Deploying on Amazon EKS *(on page 124)*.

Advantages of deploying on Azure Kubernetes Service (AKS)

You can use Azure AKS to deploy, scale up, scale down and manage containers in the cluster environment. Use the IBM® Workload Scheduler Helm chart and container images to deploy the server, console and dynamic agent to the Azure AKS public cloud. Azure AKS gives you access to helpful services. For example, you can use the Azure SQL database, a highly scalable cloud database service. See Deploying on Azure AKS *(on page 124)* for more details.

Advantages of deploying on Google GKE

Google Kubernetes Engine (GKE) provides a managed environment for deploying, managing, and scaling your containerized applications using Google infrastructure. The Google GKE environment consists of multiple machines grouped together to form a cluster. You can also deploy and run Google Cloud SQL for SQL server.

Google GKE supports session affinity in a load balancing cluster, a feature which maintains each user session always active on the same pod. This ensures that the Dynamic Workload Console always connects to the same server during a session and that the user can perform any number of operations smoothly and seamlessly.

For more information, see Deploying on Google GKE *(on page 125)*.

Chapter 1. Installing from the command-line interface

Install IBM Workload Scheduler from the command-line interface based on a typical installation scenario. Variations to the typical scenario are described in the FAQ sections.

Before you get started, download the installation images and verify the prerequisites, as described in sections Downloading installation images *(on page 31)* and Prerequisites *(on page 31)*.

Downloading installation images

Steps to take when downloading images on your workstation.

You can download installation images from IBM Fix Central.

1. Ensure that your workstation has sufficient space to store the compressed file containing the installation images. For more information about system requirements, see IBM Workload Scheduler Detailed System Requirements.
2. From IBM Fix Central, download the compressed file, containing the latest product image, to a temporary directory.
3. Extract the installation image from the downloaded file and verify that the installation image is complete.
   Extract the content of the ZIP files into a directory, using one of the extraction tools available on your system or that can be downloaded from the internet. The tool you use must be able to keep the file permissions on the extracted files, for example, `infozip`.

   On Windows™ systems, ensure that you extract the image into a path that is not very long, otherwise, the file name might be truncated. The maximum length allowed is 255 characters.

   If you are installing on a UNIX™ operating system, run the following command:

   ```
   chmod -R 755 <imagesDir>
   ```

   Note: To extract the **.zip** file onto a Windows™ 64-bit system, ensure that the image is not located on the desktop because the Windows™ operating system extract tool might encounter a problem. Choose another directory into which to extract the product image.

   On z/OS systems, to install the Dynamic Workload Console perform the following steps:
   a. Transfer the `10.x.x-IBM-DWC-Zsystem.pax` file using the FTP protocol in binary to your USS environment.
   b. Restore the code by issuing the following command:

      ```
      pax -rf 10.x.x-IBM-DWC-Zsystem.pax
      ```

   Note: DB2 is available for download from IBM Passport Advantage only. The latest versions of WebSphere Application Server Liberty Base can be downloaded from Recommended updates for WebSphere Application Server Liberty. For further details, see the 10.2.3 Quick Start Guide available from IBM Fix Central.

Prerequisites

When installing IBM® Workload Scheduler components, consider the following prerequisites.

To produce a dynamic report that lists the supported operating systems, click Supported operating systems.

For a complete list of the correct prerequisite versions to install, open the Supported software report and click on the **Prerequisites** tab.

For a complete list of system requirements (disk spaces, temporary spaces and RAM usage), see IBM Workload Scheduler Detailed System Requirements.

WebSphere Application Server Liberty Base

The latest versions of WebSphere Application Server Liberty Base can be downloaded from Recommended updates for WebSphere Application Server Liberty. For information about WebSphere Application Server Liberty Base issues and restrictions, see Runtime environment known issues and restrictions.

Before you install IBM Workload Scheduler for the first time, you must have one of the following databases installed. The following requirements apply to the RDBMS systems:

DB2

DB2® Enterprise Server Edition

A version of DB2® is bundled with the installation image.

You can install DB2® Server and the master domain manager or Dynamic Workload Console on the same workstation, then configure the database drivers from any workstation in your environment.

You can install DB2® manually.

Oracle and Amazon RDS for Oracle

You can install Oracle in the following ways:

Oracle Enterprise Edition

The advantage of choosing Oracle Enterprise Edition is that you can implement the Oracle Partitioning feature to improve the performance of event-driven workload automation. This improves rule management performance, in particular the following queries: event_rule_instance, action_run, and operator_messages. For information about event-driven workload automation, see Running event-driven workload automation *(on page)*.

Oracle Standard Edition

Oracle Standard Edition does not include the Oracle Partitioning feature. Installing this edition does not improve the performance of event-driven workload automation.

Amazon RDS for Oracle

Amazon RDS for Oracle is a robust and convenient option for managing Oracle databases in the cloud. It handles routine database tasks such as provisioning, patching, backup, recovery, and scaling, allowing you to focus on application development.

For supported versions, see the IBM Workload Scheduler System Requirements Document at IBM Workload Scheduler Detailed System Requirements.

Note:

- When installing the product on a 64-bit library operating system, use an Oracle database on a 64-bit library.
- When upgrading:
  - If you already have an RDBMS installed and you want to upgrade it, you must upgrade it **after** you upgrade IBM Workload Scheduler.
  - Use an Oracle database on a 64-bit library when installing the product on a 64-bit library.

For information about upgrading the RDBMS, see Data maintenance *(on page)*.

MSSQL

Before you create the IBM Workload Scheduler schema on the database, you must have created the directory where the IBM Workload Scheduler table spaces will be placed when the IBM Workload Scheduler schema is created. The default is `C:\MSSQL`.

Azure SQL

A family of managed, secure, and intelligent products that use the SQL Server database engine in the Azure cloud

Google Cloud SQL for SQL server

A fully-managed database service that helps you set up, maintain, manage, and administer your relational databases on Google Cloud Platform.

Amazon RDS for MSSQL

Amazon RDS for MSSQL offers a powerful and user-friendly solution for managing MSSQL databases in the cloud. It takes care of routine tasks like provisioning, patching, backups, recovery, and scaling, so you can concentrate on developing your applications.

PostgreSQL

A powerful, open source object-relational database system, which provides reliability, feature robustness, and performance.

Scanning system prerequisites for IBM Workload Scheduler

Before installing or upgrading the product, IBM Workload Scheduler automatically runs a scan on your system.

When installing IBM Workload Scheduler using the serverinst script, the script first runs the scanner to verify system prerequisites.

Note: To ensure that the prerequisite scan process does not fail, verify that the `bc` executable is present on the local system and that it is set in the PATH environment variable. If you do not want to install the `bc` executable, you can skip the prerequisites check by using the **skipcheckprereq** parameter when running the serverinst and twsinst parameters. For more information about the `bc` executable, see bc, an arbitrary precision calculator language. For more information about installation commands, see Server components installation - serverinst script *(on page 310)* and Agent installation parameters - twsinst script *(on page 84)*.

Having an environment that meets the product system requirements ensures that an installation or upgrade succeeds without any delays or complications.

The scan verifies that:

- The operating system is supported for the product.
- On UNIX™ operating systems, the necessary product libraries are installed.
- There is enough permanent and temporary disk space to install both the product and its prerequisites.
- There is enough memory and virtual memory.

Note: The scan verifies only that the environment meets the requirements of IBM Workload Scheduler. It does not check the requirements for other components, such as DB2®.

If any of these checks fails, IBM Workload Scheduler returns an error message.

The log files for the server components are located in:

On Windows™ operating systems:

<*TWA_home*>`\logs\serverinst<`*`version_number`*`>.log`

On UNIX™ and Linux™ operating systems:

<*TWA_DATA_DIR*>`/installation/logs/serverinst<`*`version_number`*`>.log`

The log files for the Dynamic Workload Console are located in:

On Windows™ operating systems:

<*DWC_home*>`\logs\dwcinst<`*`version_number`*`>.log`

On UNIX™ and Linux™ operating systems:

`<DWC_DATA_dir>/installation/logs/dwcinst<version_number>.log`

The log files for the agents are located in:

On Windows™ operating systems:

`<TWA_home>\logs\twsinst<interp><user_name><version_number>.log`

On UNIX™ and Linux™ operating systems:

`<TWA_DATA_DIR>/installation/logs/twsinst<interp><user_name><version_number>.log`

You can decide to rerun the installation or upgrade without executing the prerequisite scan. If you set the **-skipcheckprereq** parameter to `true` when performing the installation, the installation script does not execute the prerequisite scan. If a problem occurs, an error is displayed, the component is installed or upgraded, but might not work. For more information about the `-skipcheckprereq` parameter in all installation scripts, see Reference *(on page 300)*.

IBM Workload Scheduler user management

The IBM Workload Scheduler user management on UNIX and Windows operating systems

Consider the following constraints and properties for the IBM Workload Scheduler user:

On Windows operating systems:

The installation process automatically creates the IBM Workload Scheduler user. If your security policies do not allow user creation during the installation process, create the user and give it the necessary right as described in Windows user domain rights and structure *(on page 34)*.

On UNIX and Linux operating systems:

Regardless of the method of installation you choose, the IBM Workload Scheduler user must be created manually before running the installation and must be enabled to login to the machine where the master domain manager is going to be installed. Use the appropriate UNIX™ and Linux™ operating system commands to create the user.

You can choose to install with the **root user** or you can perform a **no-root installation**, using a user without root privileges. Note that if you perform a no-root installation, only the user who performs the installation can use IBM Workload Scheduler. When uninstalling, use the same user who performed the installation.

.

Note: Some operating systems require that for users with a password, the password must be changed at the first login. If this is the case, for a successful installation, you will need to log in as the user and change the password for the first time.

Windows™ user domain rights and structure

If you install on Windows™ operating systems, consider the following information.

For the installation:

- You cannot have a local user and a domain user with the same name. For example, you cannot have **user1** as local user and at the same time **user1@domain1** and **domain\user1**.
- The Windows™ user performing an agent installation and the user that will own the instance of IBM Workload Scheduler must:
  - For a local IBM Workload Scheduler user, be a member of the local administrative group
  - For a domain IBM Workload Scheduler user, be a member of the domain "users" group in the domain controller and be a member of the local administrative group.

For Windows™ IBM Workload Scheduler users:

All Windows™ IBM Workload Scheduler users must have the following user permissions. They can be granted locally. Domain level policies always override local policies, so you might be required to grant the permissions from the domain:

- Act as part of the operating system
- Allow log on locally
- Impersonate a client after authentication
- Log on as a batch job
- Log on as a service
- Replace a process level token
- Adjust memory quotas for a process (available on some configurations only)

Note: These rights are granted during the installation, but you can confirm them manually.

To run IBM Workload Scheduler command lines:

On Windows operating systems with UAC disabled:

In addition to standard Windows permissions, to log on to the machine, the user must have the "Impersonate a client after authentication" permission granted. By default, this is granted just to the "Administrators" group members. This permission is required to impersonate the TWS user and access the IBM Workload Scheduler Symphony and Mailbox.

On Windows operating systems with UAC enabled:

This is the default value. The "Impersonate a client after authentication" is not available to the user, unless the cmd shell is started with "Run as administrator" permission. To run IBM Workload Scheduler command lines, the user must have "Impersonate a client after authentication" permission defined and then start the shell with the "Run as administrator" permission authenticating with its own user ID.

For the Streamlogon user:

The user must have the "logon as batch" permission to allow IBM Workload Scheduler to create the job process. In addition, you must assign to the user "Read" and "Read & execute" permission to cmd.exe. You can assign "Read" and "Read & execute" permission to cmd.exe also to the BATCH built-in group instead of to a single user.

To manage IBM Workload Scheduler agents:

The user must be in the Administrators group or must be able to perform "Run as" as **twsuser** to reset the IBM Workload Scheduler files if a recovery is needed.

Considerations for Windows™ domain controllers running Microsoft™ Active Directory

If you want to install a IBM Workload Scheduler fault-tolerant agent on workstations where users who run jobs are domain users and the domain controller is running the Microsoft™ Active Directory, decide how to install the agents and configure the domain to have thejobmon process obtain the correct information to allow the users to run jobs.

Before running a job, jobmon retrieves information about the user running the job. If the user is a domain user and the domain controller is running Microsoft™ Active Directory, whether the user information can be retrieved depends on the information in the access control list (ACL) of that user. The main jobmon process that runs the job is started as the local system account (AUTHORITY\SYSTEM), but it immediately impersonates the *TWS_user* that owns the fault-tolerant agent. This means that for jobmon to successfully launch the job, the *TWS_user* must have an access control entry (ACE) in the ACL of the user for which it is trying to retrieve information.

Perform one of the following actions:

Enable the *TWS_user* to access a set of users that run jobs

On the domain server, edit the ACL of all users that run jobs on the workstation and add an ACE for each *TWS_user*. In this case, only specified users can run the jobs submitted by jobmon.

Allow all users to run jobs submitted by jobmon by using the TWS_BYPASS_DC=TRUE system variable

Create the TWS_BYPASS_DC=TRUE system variable, with a value not null, and reboot the workstation. In this case,jobmon obtains the user information without performing the security check for the ACE in the ACL of the user. All the local and domain users can run the jobs submitted by jobmon.

Allow all users to run jobs submitted by jobmon by setting the *TWS_user* as a domain user

Set up the *TWS_user* as a Windows™ domain user and install the instance of IBM Workload Scheduler using the *TWS_user*. In this case, all authenticated users on the domain controller can access the default ACL for a domain user. Jobs can then be launched by both local and the domain users. All the local and the domain users can run the jobs submitted by jobmon.

Exclude the workstation from the security check on users ACL

On the domain server, add the host name of the workstation where the fault-tolerant agent is installed to the Pre-Windows 2000-Compatible Access Group. In this way, from a security point of view, the domain controller interacts with this workstation as if it is in a Windows™ domain that does not support Active Directory. In this case, all the local and domain users can run the jobs submitted by jobmon. In addition, the domain controller does not prevent any local or domain users from running other processes that are not controlled by IBM Workload Scheduler.

Typical installation scenario

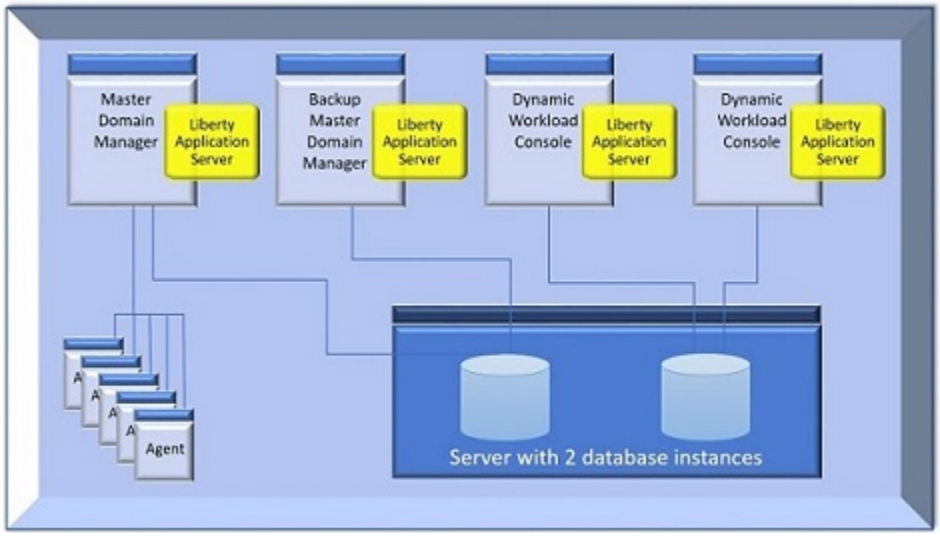
Scenario for a fresh typical installation at the latest product version of IBM® Workload Scheduler

This scenario describes how to perform a fresh install at the latest product version of the full software stack for IBM® Workload Scheduler, which consists of the following components and workstations:

- One workstation for the database server which hosts both the master domain manager and Dynamic Workload Console databases.
- One workstation for the master domain manager and the related WebSphere Application Server Liberty Base.
- One workstation for the backup master domain manager and the related WebSphere Application Server Liberty Base. The master domain manager and backup master domain manager share the same database. This ensures the backup master domain manager has the latest data and can take over seamlessly, in case the master domain manager fails.
- Two workstations for two Dynamic Workload Console installations, each of them with their related WebSphere Application Server Liberty Base. The two Dynamic Workload Console instances share the same database.
- A number of agents.

Figure 11: Typical IBM Workload Scheduler architecture *(on page 37)* describes how the IBM® Workload Scheduler components listed above are usually installed.

Figure 11. Typical IBM® Workload Scheduler architecture



Starting from version 10.2.1, using certificates is mandatory when installing or upgrading the product. You can use default certificates, generated automatically by the product with the password you specify, or you can define your own custom certificates. For more information, see Enhanced security for default certificates *(on page)*.

If you install the master domain manager on recent UNIX operating systems, you can use the OpenSSL 3.0.*x* libraries provided with the operating system. The list of UNIX operating systems whose libraries you can use is as follows:

- Ubuntu 22
- AIX 7.3
- Red Hat 9

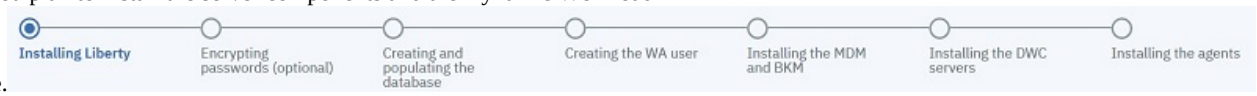
To ensure IBM® Workload Scheduler uses these libraries, always launch the installation or upgrade procedure from a brand new shell. You can also check the OpenSSL library currently in use with the which openssl command and check the OpenSSL version with the openssl version command.

This release installs a new version of the file `tws_env.sh` (`tws_env.cmd`) and also creates a backup file named, `tws_env.sh.bk` (`tws_env.cmd.bk`), which are both saved to the *TWA_HOME*/TWS directory, where *TWA_HOME* is the IBM® Workload Scheduler installation directory. After completing the installation, if you have modified the original version, merge the content of the new version with the content of the customized version to carry your customized content into the new version. When merging the two versions as described above, ensure you do not modify the paths to OpenSSL libraries.

You can now proceed to Installing WebSphere Application Server Liberty Base *(on page 37)*.

Installing WebSphere Application Server Liberty Base

WebSphere Application Server Liberty Base is required on all workstations where you plan to install the server components and the Dynamic Workload Console.



On AIX and Linux workstations, ensure you permanently set the **ulimit** parameter as follows:

- data segment process (option **-d**) = `unlimited`
- file size (option **-f**) = `unlimited`
- max user processes (option **-u**) = >`260000` up to `unlimited`
- open files (option **-n**) = >`100000` up to `unlimited`

- max memory size (option **-m**) = `unlimited`
- stack size (option **-s**) = >`33000` up to `unlimited`

On the master domain manager, these settings must be applied to:

- `root`
- the IBM® Workload Scheduler administrative user

On the Dynamic Workload Console, these settings must be applied to:

- `root`
- the Dynamic Workload Console installation user (if this user is different from root)

Ensure that your system meets the operating system and Java requirements. For more information, see WebSphere Application Server Liberty Base detailed system requirements.

You can quickly install WebSphere Application Server Liberty Base by extracting an archive file on all supported platforms.

Install WebSphere Application Server Liberty Base on all of the following workstations, which comprise a typical installation:

- master domain manager
- backup domain manager
- two Dynamic Workload Console installations on two separate workstations

If you plan to install a dynamic domain manager and its backup, these components require a separate WebSphere Application Server Liberty Base installation.

To extract the archive, you can use your own Java Ext or use the Java Ext provided with the IBM® Workload Scheduler image. The provided Java Ext is located in the following path in the image for your operating system: `<IMAGE_DIR>/TWS/<INTERP>/Tivoli_Eclipse_<INTERP>/TWS/JavaExt/`.

To install WebSphere Application Server Liberty Base, perform the following steps:

1. Find out which version of WebSphere Application Server Liberty Base is required, by running the Detailed Software Requirements report and browsing to the **Prerequisites** tab.
2. Download WebSphere Application Server Liberty Base from Recommended updates for WebSphere Application Server Liberty.
3. Install WebSphere Application Server Liberty Base by extracting the archive file to a directory of your choice.

   On Windows operating systems

   ```
   java -jar <liberty_download_dir>\wlp-base-all-<version>.jar
   --acceptLicense <install_dir>
   ```

   On UNIX operating systems

   ```
   ./java -jar <liberty_download_dir>/wlp-base-all-<version>.jar
   --acceptLicense <install_dir>
   ```

   where:

   <liberty_download_dir>

   The directory where you downloaded WebSphere Application Server Liberty Base.

   <version>

   The number of the version.

   <install_dir>

   The directory where you want to install WebSphere Application Server Liberty Base.

Note: Note that the value of the <*install_dir*> parameter must match the value to be defined for the **wlpdir** parameter when installing the master domain manager and its backup, dynamic domain manager and its backup, and the Dynamic Workload Console.

4. Ensure the IBM® Workload Scheduler administrative user has the rights to run WebSphere Application Server Liberty Base and full access to the installation directory. If WebSphere Application Server Liberty Base is shared between the master domain manager and the Dynamic Workload Console, ensure also the Dynamic Workload Console user has the same rights.

You have now successfully installed WebSphere Application Server Liberty Base.

You can now proceed to Encrypting passwords (optional) *(on page 39)*.

Encrypting passwords (optional)

How to encrypt passwords required by the installation, upgrade, and management processes.

Optional. Specifies the custom passphrase that is used to generate the key with which the command encrypts the password. If you set this parameter, inform the user who installs IBM Workload Automation that they must define the **SECUREWRAP_PASSPHRASE** environment variable in the same shell from which they run the installation command, and set it to the same value as the **passphrase** argument. On Windows operating systems, the passphrase must be at least 8 characters long.

3. Provide both the encrypted password and custom passphrase to the user in charge of installing IBM Workload Automation. You can use encrypted passwords only in association with the specific passphrase used to encrypt them.

Installing with the encrypted password

The user in charge of installing IBM Workload Automation must set the **SECUREWRAP_PASSPHRASE** environment variable by performing the following steps:

1. Open a brand new shell session.
2. Ensure that no value is set for the **SECUREWRAP_PASSPHRASE** environment variable.
3. Define the **SECUREWRAP_PASSPHRASE** environment variable and set it to the passphrase defined by the user who ran the secure command, as follows:

   ```
   SECUREWRAP_PASSPHRASE=<passphrase>
   ```

   You can use encrypted passwords only in association with the specific passphrase used to encrypt them.
4. In the same shell session, provide the encrypted passwords when running any command that uses a password. An encrypted password looks like the following example:

   ```
   {aes}AFC3jj9cROYyqR+3CONBzVi8deLb2Bossb9GGroh8UmDPGikIkzXZzid3nzY0IhnSg=
   ```

You can now proceed to Creating and populating the database *(on page 40)*.

Creating and populating the database

Create the required databases before you begin the installation.

Before you start the installation, you must create and populate the database for both the master domain manager and the Dynamic Workload Console. You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in FAQ - Database customizations *(on page 60)*. Links to customization options which are specific for a single database, if any, are provided in the related scenario.

The procedure differs for each supported database, as listed below:

DB2

- Creating and populating the database for DB2 for the master domain manager *(on page 41)*.
- Creating and populating the database for DB2 for the Dynamic Workload Console *(on page 43)*
- Creating and populating the database for DB2 for z/OS for the Dynamic Workload Console *(on page 45)*

Oracle

- Creating the database for Oracle and Amazon RDS for Oracle for the master domain manager *(on page 47)*
- Creating the database for Oracle and Amazon RDS for Oracle for the Dynamic Workload Console *(on page 49)*

MSSQL

- Creating the database for MSSQL for the master domain manager *(on page 50)*
- Creating and populating the database for MSSQL for the Dynamic Workload Console *(on page 53)*

MSSQL cloud-based databases

- Creating the database for MSSQL cloud-based databases for the master domain manager *(on page 54)*
- Creating and populating the database for MSSQL cloud-based databases for the Dynamic Workload Console *(on page 56)*

PostgreSQL

- Creating and populating the database for PostgreSQL for the master domain manager *(on page 57)*
- Creating and populating the database for PostgreSQL for the Dynamic Workload Console *(on page 58)*

A set of scripts and SQL files is provided for each database type to perform actions such as granting rights or reorganizing the database. These files are located in *inst_dir*`/TWS/dbtools` into a separate folder for each database type. To use these files, copy the relevant folder to the database server.

On UNIX™ operating systems, ensure the database administrator has read and write privileges for the IBM® Workload Scheduler installation path.

Note: If you create the schema on your own, ensure the COLLATE value is set appropriately. Consider the following examples:

DB2

```
db2 get db cfg for TWS | grep -i collating
```

The expected values are:

```
Database collating sequence = IDENTITY
Alternate collating sequence (ALT_COLLATE) =
```

MSSQL

```
select DATABASEPROPERTYEX('Your DB Name','collation')
```

The expected values is:

```
Latin1_General_BIN2
```

During database upgrade from Db2 V9.5 or earlier, the **CUR_COMMIT** configuration parameter is set to `DISABLED` to maintain the same behavior as in previous releases. For the proper functioning of IBM® Workload Scheduler and to prevent possible internal deadlocks, set the **CUR_COMMIT** parameter to `ON`. For more information, see IWS composer performance issue: Ensuring that CUR_COMMIT is ON on DB2.

Creating and populating the database for DB2 for the master domain manager

Instructions for creating and populating the IBM® Workload Scheduler database for DB2 for the master domain manager

You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in FAQ - Database customizations *(on page 60)*.

For more information about all parameters and supported values of the configureDb command, see Database configuration - configureDb script *(on page 301)*.

DB2 requires a specific procedure in which you first create the database and then create and populate the database tables. To simplify the database creation, a customized SQL file named `create_database.sql` is provided, containing the specifics for creating the IBM® Workload Scheduler database. The database administrator can use this file to create the database. After the database has been created, you can proceed to create and populate the database tables.

You can optionally configure DB2 in SSL mode on UNIX operating systems by specifying the **sslkeysfolder** and **sslpassword** parameters when you run the configureDb command. For more information, see How can I use certificates when Db2 or PostgreSQL is in SSL mode? *(on page 66)*.

You can run the configureDb command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

Default values are stored in the `configureDb<database_vendor>.properties` file, located in `image_location/TWS/interp_name`. For an example of a properties file, see What is the content of a database properties file? *(on page 66)*.

If you need to modify any of the default values, edit the `configureDb<database_vendor>.properties` file, but do not modify the `configureDb<database_vendor>.template` file located in the same path.

To create and populate the IBM® Workload Scheduler database and tables, perform the following steps:

1. On the workstation where you plan to install the master domain manager, extract the IBM® Workload Scheduler package to a directory of your choice.
2. Browse to the `image_location/TWS/interp_name/Tivoli_MDM_interp_name/TWS/tws_tools` path.
3. Edit the `create_database.sql` file by replacing the default value for the database name (**TWS**) with the name you intend to use.
4. Provide the `create_database.sql` file to the DB2 administrator to run on the DB2 database.
   The following command creates the IBM® Workload Scheduler database:

   ```
   db2 -tvf file_location>/create_database.sql
   ```

5. Instruct the DB2 administrator to create the DB2 user on the server hosting the DB2 database. You will then specify this user with the `dbuser` parameter when creating and populating the database with the configureDb command on the master domain manager.
6. Browse to the path `image_location/TWS/interp_name`.
7. Type the following command to create and populate the IBM® Workload Scheduler database tables with typical settings:

   On Windows operating systems

   ```
   cscript configureDb.vbs --rdbmstype DB2 --dbhostname DB_hostname
   --dbport db_port --dbname db_name --dbuser db_user
   --dbadminuser DB_administrator
   --dbadminuserpw DB_administrator_password
   ```

   On UNIX operating systems

   ```
   ./configureDb.sh --rdbmstype DB2 --dbhostname DB_hostname
   --dbport db_port --dbname db_name --dbuser db_user
   --dbadminuser DB_administrator
   --dbadminuserpw DB_administrator_password
   ```

   where:

   --rdbmstype

   The database vendor.

   --dbhostname *db_hostname*

   The host name or IP address of database server.

   --dbport *db_port*

   The port of the database server.

   --dbname *db_name*

   The name of the IBM® Workload Scheduler database.

--dbuser ***db_user***

The database user you must create before running the `configureDb` command.

--dbadminuser ***db_admin_user***

The database administrator user that creates the IBM® Workload Scheduler schema objects on the database server.

--dbadminuserpw ***db_admin_password***

The password of the DB administrator user that creates the IBM® Workload Scheduler schema objects on the database server. Special characters are not supported.

Note: The following parameters are also required when installing the master components and their values must be the same:

- **--rdbmstype**
- **--dbhostname**
- **--dbport**
- **--dbname**
- **--dbuser**

You have now successfully created and populated the IBM® Workload Scheduler database.

You can now proceed to Creating and populating the database for DB2 for the Dynamic Workload Console *(on page 43)*.

Creating and populating the database for DB2 for the Dynamic Workload Console

Instructions for creating and populating the Dynamic Workload Console database for DB2

Ensure a DB2 database is installed.

You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in FAQ - Database customizations *(on page 60)*.

DB2 requires a specific procedure in which you first create the database and then create and populate the database tables. To simplify the database creation, a customized SQL file named `create_database.sql` is provided containing the specifics for creating the Dynamic Workload Console database. The database administrator can use this file to create the database. After the database has been created, you can proceed to create and populate the database tables.

You can optionally configure DB2 in SSL mode on UNIX operating systems by specifying the **sslkeysfolder** and **sslpassword** parameters when you run the configureDb command. For more information, see How can I use certificates when Db2 or PostgreSQL is in SSL mode? *(on page 66)*.

You can run the configureDb command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

For more information about all parameters and supported values of the configureDb command, see Database configuration - configureDb script *(on page 301)*.

Default values are stored in the `configureDb.properties` file, located in *`image_location`*. If you need to modify any of the default values, edit the `configureDb.properties` file, but do not modify the `configureDb.template` file located in the same path.

For an example of a properties file, see What is the content of a database properties file? *(on page 66)*.

To create and populate the Dynamic Workload Console database and schema for DB2, perform the following steps:

1. On the workstation where you plan to install the Dynamic Workload Console, extract the Dynamic Workload Console package to a directory of your choice.
2. Browse to the *image_location*/DWC_*interp_name*/tools path.
3. Edit the `create_database.sql` file by replacing the default value for the database name (**DWC**) with the name you intend to use.
4. Provide the `create_database.sql` file to the DB2 administrator to run on the DB2 database.
   The following command creates the Dynamic Workload Console database:

   ```
   db2 -tvf <file_location>/create_database.sql
   ```

5. Instruct the DB2 administrator to create the DB2 user on the server hosting the DB2 database. You will then specify this user with the `dbuser` parameter when creating and populating the database with the configureDb command on the Dynamic Workload Console. When you run the configureDb command, this user is automatically granted access to the Dynamic Workload Console tables on the database server.
6. On the server where you plan to install the Dynamic Workload Console,browse to *image_location*/DWC_*interp_name*.
7. Type the following command to create and populate the Dynamic Workload Console database tables with typical settings:

   On Windows operating systems

   ```
   cscript configureDb.vbs --rdbmstype DB2 --dbhostname DB_hostname
           --dbport db_port --dbname db_name --dbuser db_user
           --dbadminuser DB_admin_user --dbadminuserpw DB_admin_pwd
   ```

   On UNIX operating systems

   ```
   ./configureDb.sh --rdbmstype DB2 --dbhostname DB_hostname
           --dbport db_port --dbname db_name --dbuser db_user
           --dbadminuser DB_admin_user --dbadminuserpw DB_admin_pwd
   ```

   where:

   --rdbmstype

   The database vendor.

   --dbhostname *db_hostname*

   The host name or IP address of database server.

   --dbport *db_port*

   The port of the database server.

   --dbname *db_name*

   The name of the Dynamic Workload Console database.

   --dbuser *db_user*

   The database user you must create before running the configureDb command. When you run the configureDb command, this user is automatically granted access to the Dynamic Workload Console tables on the database server.

   --dbadminuser *db_admin_user*

   The database administrator user that creates the Dynamic Workload Console schema objects on the database server.

   --dbadminuserpw *db_admin_password*

   The password of the DB administrator user that creates the Dynamic Workload Console schema objects on the database server. Special characters are not supported.

   Note: The following parameters specified with the configureDb command are also required when installing theDynamic Workload Console and their values must be the same:
   - **--rdbmstype**
   - **--dbhostname**
   - **--dbport**

- **--dbname**
- **--dbuser**

You have now successfully created and populated the Dynamic Workload Console database.

For more information about all parameters and supported values of the configureDb command, see Database configuration - configureDb script *(on page 301)*.

You can now proceed to Creating the IBM Workload Scheduler administrative user *(on page 69)*.

Creating and populating the database for DB2 for z/OS for the Dynamic Workload Console

Instructions for creating and populating the database for DB2 for z/OS for Dynamic Workload Console

Ensure a DB2 for z/OS database is installed.

You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in FAQ - Database customizations *(on page 60)*.

DB2 for z/OS requires a specific procedure in which you first create the database and then create and populate the database tables. To simplify the database creation, a sample JCL named `EQQINDWC` is provided with APAR PH22448 containing the specifics for creating the Dynamic Workload Console database. The database administrator can use this file to create the database. After the database has been created, you can proceed to create and populate the database tables.

You can run the **configureDb** command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

For more information about all parameters and supported values of the `configureDb` command, see Database configuration - configureDb script *(on page 301)*.

If you need to modify any of the default values, edit the `configureDb<database_vendor>.properties` file, but do not modify the `configureDb<database_vendor>.template` file located in the same path.

To create and populate the Dynamic Workload Console database and schema for DB2 for z/OS, perform the following steps:

1. From the SEQQSAMP library, edit the `EQQINDWC` sample JCL as required.

Note: The `EQQINDWC` sample JCL is provided with the APAR PH22448. If you did not install this APAR, create a JCL named `EQQINDWC` that looks like the following example:

```
//JOBCARD
//*******************************************************************/
//*                                                                 */
//*  SECURITY CLASSIFICATION:                                       */
//*  Licensed Materials - Property of HCL 5698-T08                  */
//*  Copyright HCL Technologies Ltd. 2020 All rights reserved.      */
//*  US Government Users Restricted Rights - Use, duplication       */
//*  or disclosure restricted by GSA ADP Schedule Contract          */
//*                                                                 */
//* CREATES DB2 STORAGE GROUP AND DATABASE for DWC                  */
//* NOTE1:You must tailor this JCL sample to conform to             */
//*       installation standards defined at your location.          */
//*       - Add a JOB card                                          */
//*       - Change following DB/2 values according to your          */
//*         current environment:                                    */
//*         - DSN.V11R1M0.SDSNLOAD      DB/2 library                */
//*         - DSN111.RUNLIB.LOAD        DB/2 run library            */
//*         - DBB1                      DB/2 system name            */
//*         - DSNTIA11                  DB/2 DSNTIAD plan name      */
//*         - volname                   volume name                 */
```

```
//*         - catname                  catalog name                 */
//*       - Change all the occurrences of                           */
//*         TWSSDWC if you need a storage group with a different name*/
//*                                                                  */
//* Flag Reason   Rlse   Date   Origin Flag Description              */
//* ---- -------- ------ ------ ------ ---------------------------  */
//* $EGE=PH22448  950    200121 ZLIB: DB2 on zLiberty                */
//* $ETA=PH53936  101 220418 MR: EQQINDWC MEMBER OF SEQQSAMP FOR     */
//*                              CREATION OF DB2 DATABASE FOR        */
//*                              DWCFAILS FOR DB2 V12R1M504 OR       */
//*                              higher levels                       */
//*******************************************************************/
//EQQINDWC EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB  DD  DISP=SHR,DSN=DSN.V11R1M0.SDSNLOAD
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  *
  DSN SYSTEM(DBB1)
  RUN PROGRAM(DSNTIAD) PLAN(DSNTIA11) LIB('DSN111.RUNLIB.LOAD')
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSIN    DD *
SET CURRENT APPLICATION COMPATIBILITY = 'V10R1';
CREATE STOGROUP TWSSDWC VOLUMES(volname) VCAT catname;
CREATE DATABASE DWC
BUFFERPOOL BP0
INDEXBP BP16K0
STOGROUP TWSSDWC
CCSID UNICODE;
COMMIT;
```

2. Instruct the DB2 for z/OS administrator to create the DB2 for z/OS user on the server hosting the DB2 for z/OS database. You will then specify this user with the `dbuser` parameter when creating and populating the database with the `configureDb` command on the Dynamic Workload Console. When you run the `configureDb` command, this user is automatically granted access to the Dynamic Workload Console tables on the database server.
3. On the server where you plan to install the Dynamic Workload Console, browse to the directory where you extracted the Dynamic Workload Console image.
4. Type the following command to create and populate the Dynamic Workload Console database tables with typical settings:

   On Windows operating systems

   ```
   cscript configureDb.vbs --rdbmstype DB2Z --dbhostname DB_hostname
   --dbport db_port --dbname db_name --dbuser db_user
   --dbadminuser DB_admin_user --dbadminuserpw DB_admin_pwd
   --zlocationname zOS_location_containing_db --zbufferpoolname buffer_pool_in_zOS_location
   ```

   On UNIX operating systems

   ```
   ./configureDb.sh --rdbmstype DB2Z --dbhostname DB_hostname
   --dbport db_port --dbname db_name --dbuser db_user
   --dbadminuser DB_admin_user --dbadminuserpw DB_admin_pwd
   --zlocationname zOS_location_containing_db --zbufferpoolname buffer_pool_in_zOS_location
   ```

   On z/OS operating systems

   ```
   ./configureDb.sh --rdbmstype DB2Z --dbhostname DB_hostname
   --dbport db_port --dbname db_name --dbuser db_user
   --dbadminuser DB_admin_user --dbadminuserpw DB_admin_pwd
   --zlocationname zOS_location_containing_db --zbufferpoolname buffer_pool_in_zOS_location
   ```

   where:

   --rdbmstype

   The database vendor.

   --dbhostname *db_hostname*

   The host name or IP address of database server.

   --dbport *db_port*

   The port of the database server.

--dbname ***db_name***

The name of the Dynamic Workload Console database.

--dbuser ***db_user***

The database user you must create before running the `configureDb` command. When you run the `configureDb` command, this user is automatically granted access to the Dynamic Workload Console tables on the database server.

--dbadminuser ***db_admin_user***

The database administrator user that creates the Dynamic Workload Console schema objects on the database server.

--dbadminuserpw ***db_admin_password***

The password of the DB administrator user that creates the Dynamic Workload Console schema objects on the database server. Special characters are not supported.

--zlocationname ***zos_location_containing_db***

The name of an already existing location in the z/OS environment that will contain the new database. The default value is LOC1.

--zbufferpoolname ***buffer_pool_in zos_location***

The name of an already existing buffer pool created in the location specified by `-zlocationname`. The default value is BP32K.

Note: The following parameters specified with the **configureDb** command are also required when installing the Dynamic Workload Console and their values must be the same:

- **--rdbmstype**
- **--dbhostname**
- **--dbport**
- **--dbname**
- **--dbuser**
- **--zlocationname**

You have now successfully created and populated the Dynamic Workload Console database.

You can now proceed to Creating the IBM Workload Scheduler administrative user *(on page 69)*.

Creating the database for Oracle and Amazon RDS for Oracle for the master domain manager

Instructions for creating and populating the IBM® Workload Scheduler database for Oracle and Amazon RDS for Oracle for the master domain manager

Ensure the following required tablespaces have been already created on the Oracle database server which hosts the master domain manager database:

- tablespace for IBM® Workload Scheduler data
- tablespace for IBM® Workload Scheduler log
- tablespace for IBM® Workload Scheduler plan

You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in FAQ - Database customizations *(on page 60)*.

You can run the configureDb command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

For more information about all parameters and supported values of the configureDb command, see Database configuration - configureDb script *(on page 301)*.

Default values are stored in the `configureDbOracle.properties` file, located in *`image_location`*`/TWS/`*`interp_name`*.

If you need to modify any of the default values, edit the `configureDbOracle.properties` file, but do not modify the `configureDbOracle.template` file located in the same path. For an example of a properties file, see What is the content of a database properties file? *(on page 66)*.

If you plan to use different locales in your environment, ensure you read the TWS: missing JobStream dependencies and jobs in the model and/or JobStream in the plan with Oracle Db tech note.

To create and populate the IBM® Workload Scheduler database and schema, perform the following steps:

1. On the server where you plan to install the master domain manager, extract the IBM® Workload Scheduler package to a directory of your choice.
2. Browse to *`image_location`*`/TWS/`*`interp_name`*.
3. Type the following command to create and populate the IBM® Workload Scheduler database with typical settings:

   On Windows operating systems

   ```
   cscript configureDb.vbs --rdbmstype ORACLE --dbname service_name
   --dbuser db_user --dbpassword DB_password --dbhostname DB_hostname
   --dbadminuser DB_administrator --dbadminuserpw DB_administrator_password
   --iwstsname USERS --iwslogtsname USERS --iwsplantsname USERS
   ```

   On UNIX operating systems

   ```
   ./configureDb.sh --rdbmstype ORACLE --dbname service_name
   --dbuser db_user --dbpassword DB_password --dbhostname DB_hostname
   --dbadminuser DB_administrator --dbadminuserpw DB_administrator_password
   --iwstsname USERS --iwslogtsname USERS --iwsplantsname USERS
   ```

   where:

   --rdbmstype

   The database vendor.

   --dbhostname *db_hostname*

   The host name or IP address of database server.

   --dbport *db_port*

   The port of the database server.

   --dbname *db_name*

   The service name of the IBM® Workload Scheduler database.

   dbuser *db_user*

   The user to be granted access to the IBM® Workload Scheduler tables on the database server.

   --dbpassword *db_password*

   The password for the user that has been granted access to the IBM® Workload Scheduler tables on the database server. Special characters are not supported.

   --dbadminuser*db_admin_user*

   The database administrator user that creates the IBM® Workload Scheduler schema objects on the database server.

   --dbadminuserpw *db_admin_password*

The password of the DB administrator user that creates the IBM® Workload Scheduler schema objects on the database server. Special characters are not supported.

--iwstsname|-tn ***table_space_name***

The name of the tablespace for IBM® Workload Scheduler data. This parameter is required.

--iwslogtsname|-ln ***log_table_space***

The name of the tablespace for IBM® Workload Scheduler log. This parameter is required.

--iwsplantsname|-pn ***plan_table_space***

The name of the tablespace for IBM® Workload Scheduler plan. This parameter is required.

Note: The following parameters specified with the configureDb command are also required when installing the server components and their values must be the same:

- **rdbmstype**
- **dbhostname**
- **dbport**
- **dbname**
- **dbuser**
- **dbpassword**

You have now successfully created and populated the IBM® Workload Scheduler database.

You can now proceed to Creating the database for Oracle and Amazon RDS for Oracle for the Dynamic Workload Console *(on page 49)*.

Creating the database for Oracle and Amazon RDS for Oracle for the Dynamic Workload Console

Instructions for creating and populating the Dynamic Workload Console database for Oracle and Amazon RDS for Oracle

Ensure the required tablespace for Dynamic Workload Console data has been already created on the Oracle database server which hosts the Dynamic Workload Console database.

You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in FAQ - Database customizations *(on page 60)*.

You can run the configureDb command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

For more information about all parameters and supported values of the configureDb command, see Database configuration - configureDb script *(on page 301)*.

Default values are stored in the `configureDbOracle.properties` file, located in *`image_location`*. If you need to modify any of the default values, edit the `configureDbOracle.properties` file, but do not modify the `configureDbOracle.template` file located in the same path.

To create and populate the Dynamic Workload Console database, perform the following steps:

1. On the server where you plan to install the Dynamic Workload Console, extract the Dynamic Workload Console package to a directory of your choice.
2. Browse to the directory where you extracted the package.
3. Type the following command to populate the Dynamic Workload Console database with typical settings:

   On Windows operating systems

```
cscript configureDb.vbs --rdbmstype ORACLE --dbname service_name
--dbuser db_user --dbpassword DB_password --dbhostname DB_hostname
--dbadminuser DB_administrator --dbadminuserpw DB_administrator_password
--iwstsname USERS
```

On UNIX operating systems

```
./configureDb.sh --rdbmstype ORACLE --dbname service_name
--dbuser db_user --dbpassword DB_password --dbhostname DB_hostname
--dbadminuser DB_administrator --dbadminuserpw DB_administrator_password
--iwstsname USERS
```

where:

--rdbmstype

The database vendor.

--dbname *db_name*

The service name of the Dynamic Workload Console database.

dbuser *db_user*

The user to be granted access to the Dynamic Workload Console tables on the database server.

--dbpassword *db_password*

The password for the user that has been granted access to the Dynamic Workload Console tables on the database server. Special characters are not supported.

--dbhostname *db_hostname*

The host name or IP address of database server.

--dbadminuser *db_admin_user*

The database administrator user that creates the Dynamic Workload Console schema objects on the database server.

--dbadminuserpw *db_admin_password*

The password of the DB administrator user that creates the Dynamic Workload Console schema objects on the database server. Special characters are not supported.

--iwstsname|-tn *table_space_name*

The name of the tablespace for Dynamic Workload Console data. This parameter is required.

Note: The following parameters specified with the configureDb command are also required when installing the Dynamic Workload Console and their values must be the same:

- **rdbmstype**
- **dbhostname**
- **dbport**
- **dbname**
- **dbuser**
- **dbpassword**

You have now successfully created and populated the Dynamic Workload Console database.

You can now proceed to Creating the IBM Workload Scheduler administrative user *(on page 69)*.

Creating the database for MSSQL for the master domain manager

Instructions for creating and populating the IBM® Workload Scheduler database for MSSQL for the master domain manager

You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in FAQ - Database customizations *(on page 60)*.

You can run the configureDb command specifying a typical set of parameters. In this case, default values are used for all remaining parameters. By default, MSSQL authentication is used. To modify the authentication type, see How can I specify the authentication type when using an MSSQL database? *(on page 64)*.

For more information about all parameters and supported values of the configureDb command, see Database configuration - configureDb script *(on page 301)*.

Default values are stored in the `configureDbMSSQL.properties` file, located in *`image_location`*`/TWS/`*`interp_name`*.

If you need to modify any of the default values, edit the `configureDbMSSQL.properties` file, but do not modify the `configureDbMSSQL.template` file located in the same path. For an example of a properties file, see What is the content of a database properties file? *(on page 66)*.

Note: Only on Windows systems hosting an MSSQL database, the path hosting the tablespaces must be existing before you run the configureDb.vbs command.

To create the IBM® Workload Scheduler database and schema, perform the following steps:

1. Only on Windows systems hosting an MSSQL database, create the path for hosting the following tablespaces, if the path is not already existing:
   - TWS_DATA
   - TWS_LOG
   - TWS_PLAN
2. Only on Windows systems hosting an MSSQL database, specify the path for the tablespaces when running the configureDb.vbs command or when filling in the `configureDbMSSQL.properties` properties file with the following parameters:
   - --iwstspath
   - --iwslogtspath
   - --iwsplantspath
3. On the server where you plan to install the master domain manager, extract the IBM® Workload Scheduler package to a directory of your choice.
4. Browse to *`image_location`*`/TWS/`*`interp_name`*.
5. To populate the IBM® Workload Scheduler database with typical settings, type the following command:

   On Windows operating systems

   ```
   cscript configureDb.vbs --rdbmstype MSSQL --dbname db_name
   --dbhostname db_hostname --dbadminuser db_administrator
   --dbadminuserpw db_administrator_password
   --iwstspath DATA_tablespace_path
   --iwslogtspath LOG_tablespace_path
   --iwsplantspath PLAN_tablespace_path
   ```

   On UNIX operating systems

   ```
   ./configureDb.sh --rdbmstype MSSQL --dbname db_name
   --dbhostname db_hostname --dbadminuser db_administrator
   --dbadminuserpw DB_administrator_password
   --iwstspath DATA_tablespace_path
   --iwslogtspath LOG_tablespace_path
   --iwsplantspath PLAN_tablespace_path
   ```

   where:

   --rdbmstype

   The database vendor.

   --dbhostname *db_hostname*

   The host name or IP address of database server.

--dbport ***db_port***

The port of the database server.

--dbname ***db_name***

The name of the IBM® Workload Scheduler database.

dbuser ***db_user***

The user to be granted access to the IBM® Workload Scheduler tables on the database server.

--dbadminuser ***db_admin_user***

The database administrator user that creates the IBM® Workload Scheduler schema objects on the database server.

--dbadminuserpw ***db_admin_password***

The password of the DB administrator user that creates the IBM® Workload Scheduler schema objects on the database server. Special characters are not supported.

--iwstspath|-tp ***table_space***

The path of the tablespace for IBM® Workload Scheduler or Dynamic Workload Console data. This parameter is optional. The default value for all databases other than Oracle is:

For all operating systems, except z/OS

TWS_DATA

For z/OS operating system

TWSDATA

Only on Windows systems hosting an MSSQL database, ensure the folder for the tablespace is already existing before running the configureDb command and specify the path using this parameter. Specify the path using forward slashes (/), for example: `c:/<my_path>/TWS_DATA`.

--iwslogtspath|-lp ***log_path_table_space***

The path of the tablespace for IBM® Workload Scheduler log. This parameter is optional. The default value for all databases other than Oracle is **TWS_LOG**. This parameter applies only to the server components. Only on Windows systems hosting an MSSQL database, ensure the folder for the tablespace is already existing before running the configureDb command and specify the path using this parameter. Specify the path using forward slashes (/), for example: `c:/<my_path>/TWS_LOG`.

--iwsplantspath|-pp ***plan_path_table_space***

The path of the tablespace for IBM® Workload Scheduler plan. This parameter is optional. The default value for all databases other than Oracle is **TWS_PLAN**. This parameter applies only to the server components.

Only on Windows systems hosting an MSSQL database, ensure that the folder for the tablespace is already existing before running the configureDb command and specify the path using this parameter. Specify the path using forward slashes (/), for example: `c:/<my_path>/TWS_PLAN`.

Note: The following parameters specified with the configureDb command are also required when installing the server components and their values must be the same:

- **rdbmstype**
- **dbhostname**
- **dbport**
- **dbname**
- **dbuser**

You have now successfully created and populated the IBM® Workload Scheduler database.

You can now proceed to Creating and populating the database for MSSQL for the Dynamic Workload Console *(on page 53)*.

Creating and populating the database for MSSQL for the Dynamic Workload Console

Instructions for creating and populating the Dynamic Workload Console database for MSSQL

You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in FAQ - Database customizations *(on page 60)*.

You can run the configureDb command specifying a typical set of parameters. In this case, default values are used for all remaining parameters. By default, MSSQL authentication is used. To modify the authentication type, see How can I specify the authentication type when using an MSSQL database? *(on page 64)*.

For more information about all parameters and supported values of the configureDb command, see Database configuration - configureDb script *(on page 301)*. If you need to modify any of the default values, edit the `configureDbMSSQL.properties` file, but do not modify the `configureDbMSSQL.template` file located in the same path. For an example of a properties file, see What is the content of a database properties file? *(on page 66)*.

Default values are stored in the `configureDbMSSQL.properties` file, located in *`image_location`*.

Note: Only on Windows systems hosting an MSSQL database, the path hosting the tablespace must be existing before you run the configureDb.vbs command.

To create the Dynamic Workload Console database and schema, perform the following steps:

1. Only on Windows systems hosting an MSSQL database, create the path for hosting the following tablespace, if the path is not already existing:
   - TWS_DATA
2. Only on Windows systems hosting an MSSQL database, specify the path to the folder when running the configureDb.vbs command or when filling in the `configureDbMSSQL.properties` properties file with the following parameter:
   - --iwstspath
3. On the server where you plan to install the Dynamic Workload Console, extract the Dynamic Workload Console package to a directory of your choice.
4. To populate the Dynamic Workload Console database with typical settings, type the following command:

   On Windows operating systems

   ```
   cscript configureDb.vbs --rdbmstype MSSQL --dbname db_name
   --dbhostname db_hostname --dbadminuser db_administrator
   --dbadminuserpw db_administrator_password
   --iwstspath DATA_tablespace_path
   ```

   On UNIX operating systems

   ```
   ./configureDb.sh --rdbmstype MSSQL --dbname db_name
   --dbhostname db_hostname --dbadminuser db_administrator
   --dbadminuserpw db_administrator_password
   --iwstspath DATA_tablespace_path
   ```

   where:

   --rdbmstype

   The database vendor.

   --dbname *db_name*

   The name of the Dynamic Workload Console database.

   --dbhostname *db_hostname*

   The host name or IP address of database server.

   --dbadminuser *db_admin_user*

The database administrator user that creates the Dynamic Workload Console schema objects on the database server.

--dbadminuserpw ***db_admin_password***

The password of the DB administrator user that creates the Dynamic Workload Console schema objects on the database server. Special characters are not supported.

--iwstspath|-tp ***table_space***

The path of the tablespace for IBM® Workload Scheduler or Dynamic Workload Console data. This parameter is optional. The default value for all databases other than Oracle is:

For all operating systems, except z/OS

TWS_DATA

For z/OS operating system

TWSDATA

Only on Windows systems hosting an MSSQL database, ensure the folder for the tablespace is already existing before running the configureDb command and specify the path using this parameter. Specify the path using forward slashes (/), for example: `c:/<my_path>/TWS_DATA`.

Note: The following parameters specified with the configureDb command are also required when installing the Dynamic Workload Console and their values must be the same:

- **rdbmstype**
- **dbhostname**
- **dbport**
- **dbname**

When **--rdbmstype** is set to `MSSQL`, the default value is **sa**. To install a Dynamic Workload Console with a user different from **sa**, you must create a new user in `MSSQL` and grant all the required permissions before running the configureDb command.

You have now successfully created and populated the Dynamic Workload Console database.

You can now proceed to Installing the Dynamic Workload Console *(on page 236)*.

Creating the database for MSSQL cloud-based databases for the master domain manager

Instructions for creating and populating the IBM® Workload Scheduler database for MSSQL cloud-based databases for the master domain manager.

MSSQL cloud-based databases include the following:

- Azure SQL
- Google Cloud SQL for SQL server
- Amazon RDS for MSSQL

You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in FAQ - Database customizations *(on page 60)*.

You can run the configureDb command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

For more information about all parameters and supported values of the configureDb command, see Database configuration - configureDb script *(on page 301)*.

Default values are stored in the `configureDbMSSQL.properties` file, located in *`image_location`*`/TWS/`*`interp_name`*.

If you need to modify any of the default values, edit the `configureDbMSSQL.properties` file, but do not modify the `configureDbMSSQL.template` file located in the same path. For an example of a properties file, see What is the content of a database properties file? *(on page 66)*.

To create the IBM® Workload Scheduler database and schema, perform the following steps:

1. Specify the path for the tablespaces when running the configureDb command or when filling in the `configureDbMSSQL.properties` properties file with the following parameters:
   - --iwstsname PRIMARY
   - --iwslogtsname PRIMARY
   - --iwsplantsname PRIMARY

   You can optionally modify the `PRIMARY` default values when running the configureDb command.
2. On the server where you plan to install the master domain manager, extract the IBM® Workload Scheduler package to a directory of your choice.
3. Browse to *`image_location`*`/TWS/`*`interp_name`*.
4. To populate the IBM® Workload Scheduler database with typical settings, type the following command:

   On Windows operating systems

   ```
   cscript configureDb.vbs --rdbmstype MSSQL --dbname db_name
   --dbhostname db_hostname --dbadminuser db_administrator
   --dbadminuserpw db_administrator_password
   --iwstsname DATA_tablespace_name
   --iwslogtsname LOG_tablespace_name
   --iwsplantsname PLAN_tablespace_name
   ```

   On UNIX operating systems

   ```
   ./configureDb.sh --rdbmstype MSSQL --dbname db_name
   --dbhostname db_hostname --dbadminuser db_administrator
   --dbadminuserpw DB_administrator_password
   --iwstsname DATA_tablespace_name
   --iwslogtsname LOG_tablespace_name
   --iwsplantsname PLAN_tablespace_name
   ```

   where:

   --rdbmstype

   The database vendor.

   --dbhostname *db_hostname*

   The host name or IP address of database server.

   --dbname *db_name*

   The name of the IBM® Workload Scheduler database.

   dbuser *db_user*

   The user to be granted access to the IBM® Workload Scheduler tables on the database server.

   --dbadminuser *db_admin_user*

   The database administrator user that creates the IBM® Workload Scheduler schema objects on the database server.

   --dbadminuserpw *db_admin_password*

   The password of the DB administrator user that creates the IBM® Workload Scheduler schema objects on the database server. Special characters are not supported.

   --iwstsname|-tn table_space_name

   The name of the tablespace for IBM® Workload Scheduler data.

   --iwslogtsname|-ln log_table_space

   The name of the tablespace for IBM® Workload Scheduler log.

--iwsplantsname|-pn plan_table_space

The name of the tablespace for IBM® Workload Scheduler plan.

You have now successfully created and populated the IBM® Workload Scheduler database.

You can now proceed to Creating and populating the database for MSSQL cloud-based databases for the Dynamic Workload Console *(on page 56)*.

Creating and populating the database for MSSQL cloud-based databases for the Dynamic Workload Console

Instructions for creating and populating the Dynamic Workload Console database for MSSQL cloud-based databases

MSSQL cloud-based databases include the following:

- Azure SQL
- Google Cloud SQL for SQL server
- Amazon RDS for MSSQL

You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in FAQ - Database customizations *(on page 60)*.

You can run the configureDb command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

For more information about all parameters and supported values of the configureDb command, see Database configuration - configureDb script *(on page 301)*.

If you need to modify any of the default values, edit the `configureDbMSSQL.properties` file, but do not modify the `configureDbMSSQL.template` file located in the same path. Default values are stored in the `configureDbMSSQL.properties` file, located in *`image_location`*.

For an example of a properties file, see What is the content of a database properties file? *(on page 66)*.

To create the Dynamic Workload Console database and schema, perform the following steps:

1. Specify the path to the folder when running the configureDb command or when filling in the `configureDbMSSQL.properties` properties file with the following parameter:
   - --iwstsname PRIMARY

   You can optionally modify the `PRIMARY` default value when running the configureDb command.
2. On the server where you plan to install the Dynamic Workload Console, extract the Dynamic Workload Console package to a directory of your choice.
3. To populate the Dynamic Workload Console database with typical settings, type the following command:

   On Windows operating systems

   ```
   cscript configureDb.vbs --rdbmstype MSSQL --dbname db_name
   --dbhostname db_hostname --dbadminuser db_administrator
   --dbadminuserpw db_administrator_password
   --iwstsname DATA_tablespace_name
   ```

   iwstsname *DATA_tablespace_name*

   The name of the tablespace for Dynamic Workload Console data.

Note: The following parameters specified with the **configureDb** command are also required when installing the Dynamic Workload Console and their values must be the same:
- **rdbmstype**
- **dbhostname**
- **dbport**
- **dbname**
- **dbuser**

You have now successfully created and populated the Dynamic Workload Console database.

You can now proceed to Creating the IBM Workload Scheduler administrative user *(on page 69)*.

Creating and populating the database for PostgreSQL for the master domain manager

Instructions for creating and populating the IBM® Workload Scheduler database for PostgreSQL for the master domain manager

Ensure you have performed the following tasks:

- Create the PostgreSQL database and ensure it is configured to allow remote connections.
- Create a user dedicated specifically to the new database schema and do not use the administrator user (`postgres`) for this purpose.

For more information about allowing remote connections and creating users, see the PostgreSQL documentation.

You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in FAQ - Database customizations *(on page 60)*.

For more information about all parameters and supported values of the configureDb command, see Database configuration - configureDb script *(on page 301)*.

You can optionally configure PostgreSQL in SSL mode on UNIX operating systems by specifying the **sslkeysfolder** and **sslpassword** parameters when you run the configureDb command. For more information, see How can I use certificates when Db2 or PostgreSQL is in SSL mode? *(on page 66)*

You can run the configureDb command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

Default values are stored in the `configureDbPostgresql.properties` file, located in *`image_location`*. If you need to modify any of the default values, edit the `configureDbPostgresql.properties` file, but do not modify the `configureDbPostgresql.template` file located in the same path. For an example of a properties file, see What is the content of a database properties file? *(on page 66)*.

If you need to modify any of the default values, edit the `configureDbPostgresql.properties` file, but do not modify the `configureDbPostgresql.template` file located in the same path.

To create and populate the IBM® Workload Scheduler database and tables, perform the following steps:

1. On the workstation where you plan to install the master domain manager, extract the IBM® Workload Scheduler package to a directory of your choice.
2. Browse to the path *`image_location`*`/TWS/`*`interp_name`*.
3. Type the following command to create and populate the IBM® Workload Scheduler database tables with typical settings:

   On Windows operating systems

```
cscript configureDb.vbs --rdbmstype POSTGRESQL --dbhostname DB_hostname
--dbport db_port --dbname db_name --dbuser db_user
--dbadminuser DB_administrator
--dbadminuserpw DB_administrator_password
```

On UNIX operating systems

```
./configureDb.sh --rdbmstype POSTGRESQL --dbhostname DB_hostname
--dbport db_port --dbname db_name --dbuser db_user
--dbadminuser DB_administrator
--dbadminuserpw DB_administrator_password
```

where:

--rdbmstype

The database vendor.

--dbhostname *db_hostname*

The host name or IP address of database server.

--dbport *db_port*

The port of the database server.

--dbname *db_name*

The name of the IBM® Workload Scheduler database.

--dbuser *db_user*

The database user you must create before running the `configureDb` command.

--dbadminuser *db_admin_user*

The database administrator user that creates the IBM® Workload Scheduler schema objects on the database server.

--dbadminuserpw *db_admin_password*

The password of the DB administrator user that creates the IBM® Workload Scheduler schema objects on the database server. Special characters are not supported.

Note: The following parameters are also required when installing the master components and their values must be the same:

- **--rdbmstype**
- **--dbhostname**
- **--dbport**
- **--dbname**
- **--dbuser**

You have now successfully created and populated the IBM® Workload Scheduler database.

You can now proceed to Creating and populating the database for PostgreSQL for the Dynamic Workload Console *(on page 58)*.

Creating and populating the database for PostgreSQL for the Dynamic Workload Console

Instructions for creating and populating the Dynamic Workload Console database for PostgreSQL

Ensure you have performed the following tasks:

- Create the PostgreSQL database and ensure it is configured to allow remote connections.
- Create a user dedicated specifically to the new database schema and do not use the administrator user (`postgres`) for this purpose.

For more information about allowing remote connections and creating users, see the PostgreSQL documentation.

You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in FAQ - Database customizations *(on page 60)*.

You can optionally configure PostgreSQL in SSL mode on UNIX operating systems by specifying the **sslkeysfolder** and **sslpassword** parameters when you run the configureDb command. For more information, see How can I use certificates when Db2 or PostgreSQL is in SSL mode? *(on page 66)*

You can run the configureDb command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

For more information about all parameters and supported values of the configureDb command, see Database configuration - configureDb script *(on page 301)*.

Default values are stored in the `configureDbPostgresql.properties` file, located in *`image_location`*. If you need to modify any of the default values, edit the `configureDbPostgresql.properties` file, but do not modify the `configureDbPostgresql.template` file located in the same path. For an example of a properties file, see What is the content of a database properties file? *(on page 66)*.

To create and populate the Dynamic Workload Console database and schema for PostgreSQL, perform the following steps:

1. On the workstation where you plan to install the Dynamic Workload Console, extract the Dynamic Workload Console package to a directory of your choice.
2. Browse to the path *`image_location`*`/TWS/`*`interp_name`*.
3. Type the following command to create and populate the Dynamic Workload Console database tables with typical settings:

   On Windows operating systems

   ```
   cscript configureDb.vbs --rdbmstype POSTGRESQL --dbhostname DB_hostname
   --dbport db_port --dbname db_name --dbuser db_user
   --dbadminuser DB_admin_user --dbadminuserpw DB_admin_pwd
   ```

   On UNIX operating systems

   ```
   ./configureDb.sh --rdbmstype POSTGRESQL --dbhostname DB_hostname
   --dbport db_port --dbname db_name --dbuser db_user
   --dbadminuser DB_admin_user --dbadminuserpw DB_admin_pwd
   ```

   where:

   --rdbmstype

   The database vendor.

   --dbhostname *db_hostname*

   The host name or IP address of database server.

   --dbport *db_port*

   The port of the database server.

   --dbname *db_name*

   The name of the Dynamic Workload Console database.

   --dbuser *db_user*

   The database user you must create before running the configureDb command. When you run the configureDb command, this user is automatically granted access to the Dynamic Workload Console tables on the database server.

   --dbadminuser *db_admin_user*

The database administrator user that creates the Dynamic Workload Console schema objects on the database server.

--dbadminuserpw ***db_admin_password***

The password of the DB administrator user that creates the Dynamic Workload Console schema objects on the database server. Special characters are not supported.

Note: The following parameters specified with the configureDb command are also required when installing theDynamic Workload Console and their values must be the same:

- **--rdbmstype**
- **--dbhostname**
- **--dbport**
- **--dbname**
- **--dbuser**

You have now successfully created and populated the Dynamic Workload Console database.

For more information about all parameters and supported values of the configureDb command, see Database configuration - configureDb script *(on page 301)*.

You can now proceed to Creating the IBM Workload Scheduler administrative user *(on page 69)*.

FAQ - Database customizations

A list of questions and answers related to the customization of the database:

When creating and populating a database, you might have the need to customize some parameters to suit your environment.

How can I modify the tablespace?

How can I modify the tablespace?

If you do not want to use the default tablespace name and path, you can modify them when creating and populating the database with the configureDb command.

Proceed as follows:

1. Browse to the folder containing the configureDb command. The command is located in the following path, depending on the component for which you are installing:

   master domain manager

   image_location>/TWS/*interp_name*

   Dynamic Workload Console

image_location>

2. When launching the configureDb command, as explained in Installing the master domain manager and backup master domain manager *(on page 70)* and Installing the Dynamic Workload Console servers *(on page 77)*, modify the following parameters as necessary:

   -iwstsname|-tn ***table_space_name***

   The name of the tablespace for IBM® Workload Scheduler data. This parameter is optional. The default value is **TWS_DATA**.

   -iwstspath|-tp ***table_space_path***

   The path of the tablespace for IBM® Workload Scheduler data. This parameter is optional. The default value is **TWS_DATA**.

   -iwslogtsname|-ln ***log_table_space***

   The name of the tablespace for IBM® Workload Scheduler log. This parameter is optional. The default value is **TWS_LOG**.

   -iwslogtspath|-lp ***log_path_table_space***

   The path of the tablespace for IBM® Workload Scheduler log. This parameter is optional. The default value is **TWS_LOG**.

   -iwsplantsname|-pn ***plan_table_space***

   The name of the tablespace for IBM® Workload Scheduler plan. This parameter is optional. The default value is **TWS_PLAN**.

   -iwsplantspath|-pp ***plan_path_table_space***

   The path of the tablespace for IBM® Workload Scheduler plan. This parameter is optional. The default value is **TWS_PLAN**.

For more information about the configureDb command, see Database configuration - configureDb script *(on page 301)*.

How can I avoid providing the database administrator credentials when creating the database with DB2?

Minimum required grants to manage the IBM® Workload Scheduler database with DB2

If you prefer to keep the database administrator credentials confidential and you are using DB2, you can assign a user a minimum set of grants to create, access, and modify the IBM® Workload Scheduler database.

Using the configureDb command, you can perform the following operations:

- Create the custom SQL statement to create or upgrade the IBM® Workload Scheduler database schema.
- Apply the generated SQL statement to upgrade the IBM® Workload Scheduler schema to the latest version.

Each of the previous steps requires a set of minimum grants.

Minimum required grants to create the IBM® Workload Scheduler database and table spaces

Run the configureDb command with the `--execsql` parameter set to **FALSE** to generate the `customSQLAdmin.sql` file containing the **CREATE DATABASE** statement.

After creating the database, run the configureDb command with the `--execsql` parameter set to **FALSE** to generate the `customSQL.sql` file containing the SQL statements to create table spaces and schemas. Extract from the `customSQL.sql` file the statements to **CREATE** the **BUFFERPOOLs** and **TABLESPACES**.

To create the IBM® Workload Scheduler database and the BUFFERPOOLs and TABLESPACEs, one of the following minimum grants is required:

- SYSADM
- SYSCTRL
- SELECT privilege on the PRIVILEGES administrative view

Grant to create and upgrade the IBM® Workload Scheduler database schema

To create the IBM® Workload Scheduler schema in the database, run the configureDb command with the following authorities and authorizations:

- CREATETAB on database
- CONNECT on database
- USE on all IBM® Workload Scheduler table spaces
- SELECT privilege on the PRIVILEGES administrative view

Run the configureDb command with the `--execsql` parameter set to **TRUE** to create or upgrade the IBM® Workload Scheduler database schema.

How can I configure a different temporary directory where files get downloaded?

Customizing the working directory of the database.

If you do not want to use the default working directory, where temporary files are stored, you can customize it when creating and populating the database with the configureDb command.

Proceed as follows:

1. Browse to the folder containing the configureDb command. The command is located in the following path, depending on the component for which you are installing:

   master domain manager

   image_location`/TWS/`*interp_name*

   Dynamic Workload Console

   image_location
2. When launching the configureDb command, as explained in Installing the master domain manager and backup master domain manager *(on page 70)* and Installing the Dynamic Workload Console servers *(on page 77)*, modify the following parameter as necessary:

   work_dir

   The working directory where you extract the installation image. It also contains the output produced by the command, such as the SQL statements if you set the **execsql** parameter to **false**. The default value is `/tmp` on UNIX operating systems and `C:\tmp` on Windows operating systems.

For more information about the configureDb command, see Database configuration - configureDb script *(on page 301)*.

How can I generate the SQL files required to create the database schema?

Generating the SQL files for the database schema

If you do not have the access rights to generate the schema in the database, you can create the required SQL files and then provide them to the database administrator. If you do have the access rights to generate the schema in the database, you might also want to generate the SQL files and review them before applying them to the database.

Proceed as follows:

1. Browse to the folder containing the configureDb command. The command is located in the following path, depending on the component for which you are installing:

   master domain manager

image_location/TWS/*interp_name*

Dynamic Workload Console

image_location

2. When launching the configureDb command on the workstation where you plan to install the master domain manager or Dynamic Workload Console, as explained in Creating and populating the database *(on page 40)*, set `-execsql` parameter set to **false**:

   -execsql|-es *execute_sql*

   Set to **true** to generate and run the SQL file, set to **false** to generate the SQL statement without running it. The resulting files are stored in the path defined in the **work_dir** parameter. This option is useful if you wan to review the file before running it. This parameter is optional. The default value is **true**.

3. The command creates the relevant SQL scripts containing the settings you have defined in the command line. The files are created in the working directory, which by default is `/tmp` on UNIX operating systems and `C:\tmp` on Windows operating systems.

For more information about the configureDb command, see Database configuration - configureDb script *(on page 301)*.

How can I use Oracle partitioning?

Using Oracle partitioning.

Partitioning is a powerful functionality that enables tables, indexes, and index-organized tables to be subdivided into smaller pieces, allowing these database objects to be managed and accessed at a finer level of granularity. Moreover, the Oracle partitioning feature can improve the performance of the auditing feature and event-driven workload automation. This functionality improves rule management performance, in particular the following queries:

- event_rule_instance
- action_run
- operator_messages

If partitioning is already enabled in your Oracle database, proceed as follows:

1. Browse to the folder containing the configureDb command. The command is located in the following path, depending on the component for which you are installing:

   master domain manager

   image_location/TWS/*interp_name*

   Dynamic Workload Console

   image_location

2. When launching the configureDb command, as explained in Installing the master domain manager and backup master domain manager *(on page 70)* and Installing the Dynamic Workload Console servers *(on page 77)*, modify the following parameter as necessary:

   --usePartitioning

   Only applies when installing the master domain manager. Set to **true** if you want to use the Oracle partitioning feature, otherwise set it to **false**. This parameter is optional. The default value is **true**.

For more information about the **configureDb** command, see Database configuration - configureDb script *(on page 301)*.

How can I customize the `Temp` tablespace on Oracle?

Customizing the IBM Workload Scheduler `Temp` tablespace on Oracle

If you do not want to use the default Oracle `Temp` tablespace, you can customize it when creating and populating the database with the configureDb command.

Proceed as follows:

1. Browse to the folder containing the configureDb command. The command is located in the following path, depending on the component for which you are installing:

   master domain manager

   image_location/TWS/*interp_name*

   Dynamic Workload Console

   image_location

2. When launching the configureDb command, as explained in Installing the master domain manager and backup master domain manager *(on page 70)* and Installing the Dynamic Workload Console servers *(on page 77)*, modify the following parameter:

   --iwsTempTsName *IWS_temp_path*

   Only applies when installing the master domain manager. The path of the tablespace for IBM Workload Scheduler temporary directory. This parameter is optional. The default value is **TEMP**.

For more information about the configureDb command, see Database configuration - configureDb script *(on page 301)*.

How can I check database consistency to avoid schema corruption?

Checking and maintaining database consistency

The database administrator can verify if the database schema has changed and repair any inconsistencies.

Proceed as follows:

1. Browse to the folder containing the configureDb command. The command is located in the following path, depending on the component for which you are installing:

   master domain manager

   image_location/TWS/*interp_name*

   Dynamic Workload Console

   image_location

2. When launching the configureDb command, as explained in Installing the master domain manager and backup master domain manager *(on page 70)* and Installing the Dynamic Workload Console servers *(on page 77)*, set the `-execsql` parameter to **false**:

   -execsql|-es *execute_sql*

   Set to **true** to generate and run the SQL file, set to **false** to generate the SQL statement without running it. The resulting files are stored in the path defined in the **work_dir** parameter. This option is useful if you wan to review the file before running it. This parameter is optional. The default value is **true**.

   This parameter generates a number of SQL files, which you can check to look for any inconsistencies. For example, if you find CREATE instructions, this means that some records or indexes are missing in the database.

3. If you identify any inconsistencies, provide the files to the database administrator to run on the database and fix the inconsistencies.

For more information about the configureDb command, see Database configuration - configureDb script *(on page 301)*.

How can I specify the authentication type when using an MSSQL database?

Configuring the authentication type for the MSSQL database.

When using an MSSQL database, you can choose between two different authentication types:

- MSSQL authentication. This is the default value.
- Windows authentication

To define the authentication type, proceed as follows:

1. Browse to the folder containing the configureDb command. The command is located in the following path, depending on the component for which you are installing:

   master domain manager

   image_location/TWS/*interp_name*

   Dynamic Workload Console

   image_location

2. When launching the configureDb command, as explained in Creating and populating the database *(on page 40)*, specify the **auth_type** argument with one of the following values:

   SQLSERVER

   Enables MSSQL authentication type. Only the user specified with the **--dbadminuser** argument has the grants to administer the IBM® Workload Scheduler database. This is the default value.

   WINDOWS

   Enables Windows authentication type. The Windows user you used to log on to the workstation is assigned the grants to administer the IBM® Workload Scheduler database.

For more information about all parameters and supported values of the configureDb command, see Database configuration - configureDb script *(on page 301)*.

How can I customize the JDBC drivers for the database?

How can I customize the JDBC drivers for the database?

If you do not want to use the default JDBC drivers, for example because more updated drivers have been released in the meantime, you can replace them with a few easy steps for both the master domain manager and Dynamic Workload Console.

Proceed as follows:

1. Download the updated JDBC drivers for your database.
2. Create a backup of the existing JDBC drivers installed together with the product in the following paths:

   master domain manager

   On Windows operating systems

   TWA_home\TWS\jdbcdrivers*default_RDBMS*

   On UNIX operating systems

   TWA_home/TWS/jdbcdrivers/*default_RDBMS*

   Dynamic Workload Console

   On Windows operating systems

   DWC_home\jdbcdrivers*default_RDBMS*

   On UNIX operating systems

   DWC_home/jdbcdrivers/*default_RDBMS*

   where

   default_RDBMS

   Indicates one of the following directories related to the database you are using for the master domain manager and the Dynamic Workload Console:
   - db2
   - db2z

    - mssql (applies to MSSQL and supported MSSQL cloud-based databases)
    - oracle (applies to Oracle and Amazon RDS for Oracle)
    - postgresql
3. Stop WebSphere Application Server Liberty Base for master domain manager and Dynamic Workload Console, as described in Application server - starting and stopping *(on page)*.
4. Replace the default JDBC drivers with the updated ones. Ensure you maintain the same path and rename the updated drivers to the exact name of the previous drivers.
5. Start WebSphere Application Server Liberty Base for master domain manager and Dynamic Workload Console, as described in Application server - starting and stopping *(on page)*.

Note: When you upgrade the master domain manager and Dynamic Workload Console to a new product version, the customized JDBC drivers are replaced by the drivers included in the product installation packages. To continue using custom JDBC drivers, repeat this procedure.

How can I grant access to the database when the user installing the product is not the database administrator?

Steps to grant access to the database tables when the user installing the product is not the database administrator.

If the user installing the product is not the database administrator, ensure you run the `grant_twsuser.sql` script before you run the `configureDb` script.

This ensures the database user is granted all proper rights.

The `grant_twsuser.sql` is available in *TWA_home*`/TWS/dbtools/<`*database_vendor*`>/sql`.

How can I use certificates when Db2 or PostgreSQL is in SSL mode?

How can I use certificates when Db2 or PostgreSQL is in SSL mode?

If you have Db2 or PostgreSQL set up in SSL mode on a UNIX operating system, you can add the database certificate to the existing certificates. You can use this configuration on the following components:

- master domain manager
- dynamic domain manager
- Dynamic Workload Console

Proceed as follows:

1. On the workstation where you plan to install the master domain manager, create a folder for storing the certificates.
2. Within this folder, create a subfolder named `additionalCAs`.
3. Obtain the certificates from the database administrator.
4. Store the certificates in `.crt` format in the `additionalCAs` folder.
5. Log in to the component for which you are configuring the database, as listed above.
6. Run the configureDb script as explained in Creating and populating the database for DB2 for the master domain manager *(on page 41)* and Creating and populating the database for DB2 for the Dynamic Workload Console *(on page 43)*, or in Creating and populating the database for PostgreSQL for the master domain manager *(on page 57)* and Creating and populating the database for PostgreSQL for the Dynamic Workload Console *(on page 58)*, depending on the database you are using. Ensure you use the **sslkeysfolder** and **sslkeysfolder** parameter to specify the path to the folder containing the certificates.
7. Proceed with the installation as described in Typical installation scenario *(on page 36)*.

What is the content of a database properties file?

Contents of the `configureDB.properties` file.

You can use properties files for providing input to the configureDB command without typing parameters in the command line when creating the database for the master domain manager and Dynamic Workload Console.

Consider the following example for the master domain manager database:

```
#This properties are the default for configureDb.sh command in configureDb.template file
#This properties are the input  for configureDb.sh command with -f option in configureDb.propeties file
#N.B.configureDb.template must not be changed, while configureDb.propeties can be changed when using -f
 option

#--lang language: C|en|de|es|fr|it|ja|ko|pt_BR|ru|zh_CN|zh_TW
LANG=

#--work_dir Working directory where user has write access. Used to modify input file for the db tool
 (optional, default: see below)
WORK_DIR=

#--log_dir Working directory where user has write access. Used to log (optional, default: see below)
CONFDB_LOG_DIR=

#--rdbmstype|-r The rdbmstype:      DB2 | ORACLE | MSSQL | POSTGRESQL
RDBMS_TYPE=

#--componenttype The IWS component that must be installed: MDM, BKM, DDM or BDM (default: see below)
COMPONENT_TYPE=MDM

#--dbdriverpath
DB_DRIVER_PATH=

#--dbname The name of IWS Database (default: see below)
DB_NAME=TWS

#--dbhostname The host name or IP address of DB server
DB_HOST_NAME=

#--dbport The port of the DB server
DB_PORT=50000

#--dbadminuser DB administrator user that creates the IWS schema objects on the DB server
DB_ADMIN_USER=db2admin

#--dbadminuserpw The password of the DB administrator user that creates the IWS schema objects on the DB2
 server
DB_ADMIN_USER_PWD=

#--dbuser DB user that accesses the IWS tables on the DB server
DB_USER=db2tws

#--dbpassword DB user that accesses the IWS tables on the DB server
DB_PASSWORD=

#--wlpdir|-w wlp directory needed only if any password in input is encrypted and has the form {xor}password
WLP_INSTALL_DIR=

#--iwstsname The name of the tablespace for IWS data (default: see below)
IWS_TS_NAME=TWS_DATA

#--iwstspath The path of the tablespace for IWS data (default: see below)
IWS_TS_PATH=TWS_DATA

#--iwslogtsname The name of the tablespace for IWS log (default: see below)

#--iwslogtspath The path of the tablespace for IWS log (default: see below)
IWS_LOG_TS_PATH=TWS_LOG

#--iwsplantsname The name of the tablespace for IWS plan (default: see below)
IWS_PLAN_TS_NAME=TWS_PLAN

#--iwsplantspath The path of the tablespace for IWS plan (default: see below)
IWS_PLAN_TS_PATH=TWS_PLAN

# Automatically apply the generated SQL statements needed to create the IWS database schema objects (Default:
 TRUE)
# If you want manually apply the generated statement in ./customSQL.sql file, set FALSE.
#--execsql
EXEC_GENERATED_SQL=TRUE
```

```
# --------------------------------
# needed for SSL
# --------------------------------
#  Configuration options when customized certificates are used for SSL connections:
#--sslkeysfolder        The name and path of the folder containing certificates in PEM format.
#                       This parameter is required if you set the --dbsslconnection parameter to true.
SSL_KEY_FOLDER=
#--sslpassword          If you provide PEM certificates with the --sslkeysfolder parameter, this is the
 password for the certificates automatically generated by the installation program.
SSL_PASSWORD=
# --------------------------------
# needed for SSL  unix only
# --------------------------------
#--dbsslconnection       true | false  (DB2 only)
DB_SSL_CONNECTION=false
```

Consider the following example for the Db2 database for the Dynamic Workload Console:

```
#This properties are the default for configureDb.sh command in configureDb.template file
#This properties are the input  for configureDb.sh command with -f option in configureDb.propeties file
#N.B.configureDb.template must not be changed, while configureDb.propeties can be changed when using -f
 option

#--lang language: C|en|de|es|fr|it|ja|ko|pt_BR|ru|zh_CN|zh_TW
LANG=

#--work_dir Working directory where user has write access. Used to modify input file for the db tool
 (optional, default: see below)
WORK_DIR=

#--log_dir Working directory where user has write access. Used to log (optional, default: see below)
CONFDB_LOG_DIR=

#--rdbmstype|-r The rdbmstype:      DB2 | DB2Z | ORACLE | MSSQL | POSTGRESQL
RDBMS_TYPE=DB2

#--componenttype The DWC component that must be installed: DWC
COMPONENT_TYPE=DWC

#--dbdriverpath
DB_DRIVER_PATH=

#--dbname  The name of DWC Database (default: see below)
DB_NAME=TDWC

#--dbhostname The host name or IP address of DB server
DB_HOST_NAME=<my_DB_host>

#--dbport The port of the DB server
DB_PORT=50000

#--dbadminuser DB administrator user that creates the IWS schema objects on the DB server
DB_ADMIN_USER=db2admin

#--dbadminuserpw The password of the DB administrator user that creates the DWC schema objects on the DB2
 server
DB_ADMIN_USER_PWD=<database_administrator_password>

#--dbuser DB user that accesses the DWC tables on the DB server
DB_USER=db2dwc

#--dbpassword DB user that accesses the DWC tables on the DB server
DB_PASSWORD=database_password

#--wlpdir|-w wlp directory needed only if any password in input is encrypted and has the form {xor}password
WLP_INSTALL_DIR=<WebSphere Application Server
                       Liberty_installation_directory>

#--iwstsname The name of the tablespace for data (default: TWS_DATA)
IWS_TS_NAME=TWS_DATA

#--iwstspath The path of the tablespace for data (default: TWS_DATA)
IWS_TS_PATH=TWS_DATA
```

```
# Automatically apply the generated SQL statements needed to create the DWC database schema objects (Default:
 TRUE)
# If you want manually apply the generated statement in ./customSQL.sql file, set FALSE.
#--execsql
EXEC_GENERATED_SQL=TRUE

# --------------------------------
# needed for SSL
# --------------------------------
#  Configuration options when customized certificates are used for SSL connections:
#--sslkeysfolder        The name and path of the folder containing certificates in PEM format.
#                       If you provide PEM certificates, the installation program generates the keystore and
 truststore files using the password you specify with the --sslpassword parameter.
#                       This parameter is required if you set the --dbsslconnection parameter to true.
SSL_KEY_FOLDER=
#--sslpassword          If you provide PEM certificates with the --sslkeysfolder parameter, this is the
 password for the certificates automatically generated by the installation program.
SSL_PASSWORD=
# --------------------------------
# needed for SSL  unix only
# --------------------------------
#--dbsslconnection       true | false  (DB2 only)
DB_SSL_CONNECTION=false
```

Creating the IBM® Workload Scheduler administrative user

Instructions to create the IBM® Workload Scheduler administrative user

IBM® Workload Scheduler administrative user

The IBM® Workload Scheduler administrator creates the administrative user (**wauser**). The administrative user is the user for which the product will be installed in the subsequent steps. This implies that this user has full access to all scheduling objects.

The user name can contain alphanumeric, dash (-), and underscore (_) characters; it cannot contain national characters. The first character of the user name must be a letter.

The following considerations apply:

On Windows operating systems:

- If this user account does not already exist, it is automatically created at installation time.
- If installing on a Windows™ server in a domain, do not define a domain and local ID with the same user name.
- If you specify a domain user, define the name as *domain_name**user_name*.
- If you specify a local user, define the name as *system_name**user_name*. Type and confirm the password.

On UNIX and Linux operating systems:

This user account must be created manually before running the installation and must be enabled to login to the machine where the master domain manager is going to be installed. Create a user with a home directory and group. Use the appropriate UNIX and Linux operating system commands to create the user.

Important: Group names that contain a "/" (forward slash) character can cause permissions to not be set correctly. When IBM® Workload Scheduler retrieves credentials from WebSphere Application Server Liberty, it parses the returned list of groups names assuming they are saved in the format `<realm_name>/<group_name>`. If the group name, the realm name, or both contain a "/" character, the parsing fails.

You can also install IBM® Workload Scheduler using a user different from the root user. This installation method is known as **no-root installation** and applies to all IBM® Workload Scheduler components. Note that if you

choose this installation method, only the user who performs the installation can use IBM® Workload Scheduler. For this reason, the typical installation scenario described in this section uses the root user.

For more information, see IBM Workload Scheduler user management *(on page 34)*.

You can now proceed to Installing the master domain manager and backup master domain manager *(on page 70)*.

Installing the master domain manager and backup master domain manager

A fresh installation for the master domain manager and the backup master domain manager

Note: Automatic failover triggers a switch to a backup master domain manager without manual intervention under certain conditions. To take advantage of this feature, you must install the master domain manager and backup master domain managers with the same user. With a fresh installation of a master domain manager on Linux and UNIX, a new extended agent is installed on the master domain manager workstation which is used to communicate where to run the FINAL job stream. For information about configuring automatic failover, see Automatic failover *(on page)*.

Procedure to install a master domain manager and backup master domain manager

Before starting the installation, ensure the following steps have been completed:

1. Installing WebSphere Application Server Liberty Base *(on page 37)* on the workstation where you plan to install the master domain manager and on the workstation where you plan to install the backup master domain manager.
2. Creating and populating the database *(on page 40)*
3. Creating the IBM Workload Scheduler administrative user *(on page 69)*
4. On UNIX™ operating systems, ensure that **umask** is set to **022**. To verify that **umask** is set to the correct value, from a command prompt, run the **umask** command. If the value is different from **022**, modify it by running the following command:

   ```
   umask 022
   ```

Note: When installing a backup master domain manager, the backup points to the existing IBM® Workload Scheduler database. In this case, creating and populating the database is not required.

You can perform a typical installation, as described in the following scenario, or you can customize the installation parameters, as described in FAQ - master domain manager and backup master domain manager customizations *(on page 74)*.

For more information about all **serverinst** parameters and default values, see Server components installation - serverinst script *(on page 310)*.

Certificates are now required when installing or upgrading IBM® Workload Scheduler. You can no longer install nor upgrade IBM® Workload Scheduler without securing your environment with certificates. The required certificates are:

- ca.crt
- tls.key
- tls.crt

For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).

The procedure to install the master domain manager and backup master domain manager is exactly the same, with the difference that it is performed on two different workstations and that each installation points to its local WebSphere Application Server

Liberty Base installation. IBM® Workload Scheduler determines whether or not a master domain manager is already present in the environment and proceeds to install a master domain manager or backup master domain manager accordingly.

The IBM® Workload Scheduler administrator installs the master domain manager and backup master domain manager. The following information is required:

Table 3. Required information

Required information for performing the installation

Command parameter	Information type	Provided in..
Database information		
--rdbmstype	database type	Creating and populating the database *(on page 40)*
--dbhostname	database hostname	
--dbport	database port	
--dbname	database name	
--dbuser	database user name	
--dbpassword	database password	
IBM® Workload Scheduler information		
--wauser	IBM® Workload Scheduler administrative user name	Creating the IBM Workload Scheduler administrative user *(on page 69)*
--wapassword	IBM® Workload Scheduler administrative user password	
WebSphere Application Server Liberty Base information		
--wlpdir	WebSphere Application Server Liberty Base installation directory	Installing WebSphere Application Server Liberty Base *(on page 37)*
Security information		
sslkeysfolder	name and path of the folder containing certificates	Installing the master domain manager and backup master domain manager *(on page 70)*
--sslpassword	password for the certificates	Current procedure
IBM® Workload Scheduler installation directory		
--inst_dir	installation directory	Current procedure

You can run the **serverinst** command specifying a typical set of parameters. In this case, default values are used for all remaining parameters. For more information about all **serverinst** parameters and default values, see Server components installation - serverinst script *(on page 310)*.

A properties file named `serverinst.properties` is available if you do not want to type parameters in the command line. This is expecially useful if you need to specify many parameters or if you want to reuse the file for several installations. The file is located in *`image_location`*`/TWS/`*`interp_name`*.

You can specify values in the properties file, type them in the command line, or use both methods. If a parameter is specified both in the properties file and in the command line, the command line value takes precedence.

If you need to modify any of the default values, edit the `serverinst.properties` file, but do not modify the `serverinst.template` file located in the same path.

To install the master domain manager and backup master domain manager, perform the following steps:

1. Log in to the workstation where you plan to install the master domain manager.
2. Download the installation images from IBM Fix Central.
3. Browse to the folder where the `serverinst` command is located in *`image_location`*`/TWS/`*`interp_name`*.
4. Start the installation specifying a typical set of parameters. In this case, default values are used for all remaining parameters:

   On Windows operating systems

   ```
   cscript serverinst.vbs --acceptlicense yes --rdbmstype <db_type>
   --dbhostname <db_hostname> --dbport <db_port> --dbname <db_name>
   --dbuser <db_user> --dbpassword <db_password> --wauser <wa_user>
   --wapassword <wa_password> --wlpdir <Liberty_installation_dir>\wlp
   --sslkeysfolder <certificate_files_path> --sslpassword <keystore_truststore_password>
   --inst_dir <installation_dir>
   ```

   On UNIX operating systems

   ```
   ./serverinst.sh --acceptlicense yes --rdbmstype <db_type>
   --dbhostname <db_hostname> --dbport <db_port> --dbname <db_name>
   --dbuser <db_user> --dbpassword <db_password> --wauser <wa_user>
   --wapassword <wa_password> --wlpdir <Liberty_installation_dir>/wlp
   --sslkeysfolder <certificate_files_path> --sslpassword <keystore_truststore_password>
   --inst_dir <installation_dir>
   ```

   where

   --acceptlicense

   Specify **yes** to accept the product license.

   --rdbmstype|-r *rdbms_type*

   The database type. Supported databases are:
   - **DB2**
   - **ORACLE** This value applies to Oracle and Amazon RDS for Oracle
   - **MSSQL** This value applies to MSSQL and MSSQL cloud-based databases.
   - **POSTGRESQL**

   This parameter is required and has no default value.

   --dbhostname *db_hostname*

   The host name or IP address of database server.

   --dbport *db_port*

   The port of the database server.

   --dbname *db_name*

   The name of the IBM® Workload Scheduler database.

   --dbuser *db_user*

   The database user that has been granted access to the IBM® Workload Scheduler tables on the database server.

   --dbpassword *db_password*

   The password for the user that has been granted access to the IBM® Workload Scheduler tables on the database server. Special characters are not supported.

   --wauser *user_name*

   The user for which you are installing IBM Workload Scheduler.

   --wapassword *wauser_password*

   The password of the user for which you are installing IBM Workload Scheduler.

   On Windows operating systems

   Supported characters for the password are alphanumeric, dash (-), underscore (_) characters, and ()|?*~+.@!^

   On UNIX operating systems

Supported characters for the password are any alphanumeric, dash (-), underscore (_) characters, and ()|?=*~+.

--wlpdir

The path where WebSphere Application Server Liberty Base is installed.

--sslkeysfolder ***keystore_truststore_folder***

The name and path of the folder containing certificates in PEM format. The installation program automatically processes the keystore and truststore files using the password you specify with the **--sslpassword** parameter. The folder must contain the following files:

- **ca.crt**

  The Certificate Authority (CA) public certificate. Note that if certificates being installed are part of a chain consisting of 3 or more certificates (one Root CA, followed by one or more Intermediate CAs, followed by the end user certificate), then this file must contain the Root CA certificate only. Any Intermediate CA certificates must be stored in the `additionalCAs` subfolder, which therefore becomes a mandatory subfolder. Each Intermediate CA must be stored in the `additionalCAs` subfolder in its own file.

  Note: From V10.2.3, if certificates being installed are part of a chain, the ca.crt can contain also the intermediate CAs. In this case, it must begin with one or more intermediate CA certificates and end with the Root ca.

- **tls.key**

  The private key of the end user certificate for the instance to be installed.

- **tls.crt**

  The public part of the previous key, that is the end user certificate.

For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).

You can optionally create a subfolder to contain one or more `*.crt` files to be added to the server truststore as trusted CA, whose name must be `additionalCAs`. This can be used for example to add to the list of trusted CAs the certificate of the LDAP server or DB2 server. Additionally, you can store here any intermediate CA certificate to be added to the truststore. The subfolder must be named **additionalCAs**. Note that if the end user certificate being installed in the instance is part of a chain consisting of 3 or more certificates (one Root CA, followed by one or more Intermediate CAs, followed by the end user certificate), then the Intermediate CAs certificates must be stored in the `additionalCAs` subfolder, which therefore becomes a mandatory subfolder. Each Intermediate CA must be stored in the `additionalCAs` subfolder in its own file.

For further information about how to generate custom certificates, see Managing certificates using Certman *(on page)*.

--sslpassword ***ssl_password***

The password for the custom certificates and the path to the folder containing certificates in PEM format with the **sslkeysfolder** parameter.

For more information, see sslkeysfolder *(on page 316)*.

You can optionally encrypt the password using the secure script. For more information, see Optional password encryption - secure script *(on page 300)*.

--inst_dir ***installation_dir***

The directory of the IBM Workload Scheduler installation.

Note: The values for the following parameters must match the values you provided when creating and populating the database:
- **--rdbmstype**
- **--dbhostname**
- **--dbport**
- **--dbname**
- **--dbuser**
- **--dbpassword**

Note: Before starting the deployment of a new master domain manager or backup master domain manager on an already used database, be sure that no failed plan creation/extension has been performed. If a failed plan creation or extension has been performed, resolve the failure before attempting the new deployment or unlock the database by running the planman unlock db command.

5. If you are installing a backup master domain manager, it is crucial to use the same encryption keys as those on the master domain manager, to ensure it can correctly decrypt encrypted files, such as the Symphony file. To achieve this, perform the following steps:
   a. Backup the files located in the *TWA_DATA_DIR*`\ssl\aes` folder on the backup master domain manager.
   b. Copy the files from the *TWA_DATA_DIR*`\ssl\aes` folder on the master domain manager to the *TWA_DATA_DIR*`\ssl\aes` folder on the backup master domain manager.
6. To verify that the installation completed successfully, browse to the directory where you installed the master domain manager and type the following commands:

   On UNIX operating systems

   ```
   . ./tws_env.sh
   ```

   On Windows operating systems

   ```
   tws_env.cmd
   ```

   ```
   optman ls
   ```

   This command lists the IBM® Workload Scheduler configurations settings and confirms that IBM® Workload Scheduler installed correctly.

   You can also optionally run JnextPlan -for 0000 to extend by 0 hours and 0 minutes the production plan and add into the production plan (Symphony) the newly-created workstation, or wait for the FINAL job stream to complete, then run composer list cpu=*server_workstation_name* to ensure the agents have registered. You can also run a test job to ensure everything is working correctly.

You have now successfully installed the master domain manager and backup master domain manager.

If you want to customize more installation parameters, see FAQ - master domain manager and backup master domain manager customizations *(on page 74)*.

You can proceed to Installing the Dynamic Workload Console servers *(on page 77)*.

FAQ - master domain manager and backup master domain manager customizations

A list of questions and answers related to the customization of the master domain manager and backup master domain manager installation

When installing the master domain manager and backup master domain manager, you can perform a typical installation, as described in Installing the master domain manager and backup master domain manager *(on page 70)* or you can customize a number of parameters, as described in the following scenarios:

How do I customize general information for the master domain manager installation?

How to customize general information for the master domain manager installation.

How do I define the language of the messages?

To define the language in which messages are displayed, use the **-lang** parameter, as follows:

-lang ***lang_id***

The language in which the `serverinst` messages are displayed. If not specified, the system LANG is used. If the related catalog is missing, the default C language catalog is used. If neither **-lang** nor LANG are used, the default codepage is set to SBCS. For a list of valid values for these variables, see the following table:

Table 4. Valid values for -lang and LANG parameter

Language	Value
Brazilian portuguese	pt_BR
Chinese (traditional and simplified)	zh_CN, zh_TW
English	en
French	fr
German	de
Italian	it
Japanese	ja
Korean	ko
Russian	ru
Spanish	es

Note: This is the language in which the installation log is recorded and not the language of the installed engine instance. serverinst installs all languages as default.

How do I modify the installation directory?

To modify the directory where the product is installed, use the **-inst_dir** parameter, as follows:

-inst_dir ***installation_dir***

The directory of the IBM Workload Scheduler installation. This parameter is optional. The default value is calculated at installation time, based on the user performing the installation.

-work_dir ***working_dir***

The temporary directory used by the program to deploy the installation process files. This parameter is optional. The default value is calculated at installation time, based on the user performing the installation.

I am confident that all my prerequisites are in order. How do I skip the prerequisites check?

To skip the prerequisites, use the **-skipcheckprereq** parameter, as follows:

-skipcheckprereq

If you set this parameter to `false`, IBM Workload Scheduler does not scan system prerequisites before starting the installation. This parameter is optional. The default value is `true`. For more information about the prerequisite check, see Scanning system prerequisites for IBM Workload Scheduler *(on page 33)*.

How do I customize configuration information for the data source?

How to customize configuration information for the data source used by the master domain manager

How do I change the RDBMS type?

To use a different database than the default DB2, use the **-rdbmstype** parameter when typing the **serverinst** command, as follows:

--rdbmstype|-r *rdbms_type*

The database type. Supported databases are:

- **DB2**
- **ORACLE** This value applies to Oracle and Amazon RDS for Oracle
- **MSSQL** This value applies to MSSQL and MSSQL cloud-based databases.
- **POSTGRESQL**

This parameter is required and has no default value.

For more information about supported database versions, see the Data Integration report, click on the **Prerequisites** tab, then click on **Databases**.

I prefer not to use the default IBM® Workload Scheduler database name (TWS). How do I change the database name?

To use a different database than the default DB2, use the **-dbname** parameter, as follows:

dbname *db_name*

Specify the name you want to use for the database. Note that this name must match the name specified in the `configureDb` command. For more information about the `configureDb` command, see Database configuration - configureDb script *(on page 301)*.

How can I specify a different database user?

To specify a different database name than the default value, use the **-dbuser** parameter, as follows:

dbuser *db_user*

Specify the name of the database user that accesses the IBM® Workload Scheduler tables on the database server.

How can I specify a different database port?

To specify a different database port than the default value, use the **-dbport** parameter, as follows:

dbport *db_port*

Specify the port of the database server.

How do I customize configuration information for the master domain manager?

How to customize the configuration of the master domain manager

How can I customize the *data_dir* folder to maintain the previous behavior and store the data generated by IBM Workload Scheduler, such as logs, and configuration information together with the product binaries?

By default, at installation time product data and data generated by IBM® Workload Scheduler, such as logs and configuration information are stored in the *data_dir* folder, separated from the product binaries.

If you want to revert to the previous behavior, where product data and product binaries were stored together, use the **--data_dir** argument to specify the IBM® Workload Scheduler. For more information about the **--data_dir** argument, see Server components installation - serverinst script *(on page 310)*.

You can also specify the **--data_dir** argument when installing the Dynamic Workload Console with the dwcinst command and the agents with the twsinst command. For more information, see Dynamic Workload Console installation - dwcinst script *(on page 320)* and Agent installation parameters - twsinst script *(on page 84)*.

If you deploy the product components using Docker containers, the `<data_dir>` is set to the default directory name and location, and it cannot be modified.

How do I connect a new master domain manager to an existing Dynamic Workload Console?

Share certificates between a new master domain manager and an existing Dynamic Workload Console.

If you install a new master domain manager and you want it to connect to an existing Dynamic Workload Console, you need to import the master domain manager certificates into the Dynamic Workload Console keystore. For further information about how to import certificates by using Certman, see Import certificates from a master domain manager into the Dynamic Workload Console *(on page)*.

Installing the Dynamic Workload Console servers

Procedure for installing two Dynamic Workload Console servers on two separate nodes.

Procedure for installing the Dynamic Workload Console

In this scenario, the IBM® Workload Scheduler administrator installs two Dynamic Workload Console instances on two separate workstations, sharing the same remote database. The IBM® Workload Scheduler administrator performs the operations listed below on both workstations.

Certificates are now required when installing or upgrading IBM® Workload Scheduler. You can no longer install nor upgrade IBM® Workload Scheduler without securing your environment with certificates. The required certificates are:

- ca.crt
- tls.key
- tls.crt

For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).

Note: If you are installing the Dynamic Workload Console version 10.2.3 or later, the Federator is also automatically installed. This component enables you to monitor your objects through the Orchestration Monitor page of the Dynamic Workload Console. For detailed information about how to configure and use the Federator, see Mirroring the z/OS current plan to enable the Orchestration Monitor *(on page)*.

If you are installing the on a z/OS operating system, see Installing a Dynamic Workload Console server *(on page)*.

The IBM® Workload Scheduler administrator installs the Dynamic Workload Console. The following information is required:

Table 5. Required information

Command parameter	Required information	Provided in..
Database information		
--rdbmstype	database type	Creating and populating the database *(on page 40)*
--dbhostname	database hostname	
--dbport	database port	
--dbname	database name	
--dbuser	database user name	
--dbpassword	database password	
Security information		
--sslkeysfolder	name and path of the folder containing certificates	Installing the master domain manager and backup master domain manager *(on page 70)*
--sslpassword	password for the certificates	Installing the master domain manager and backup master domain manager *(on page 70)*
WebSphere Application Server Liberty Base information		
--wlpdir	WebSphere Application Server Liberty Base installation directory	Installing WebSphere Application Server Liberty Base *(on page 37)*

You can run the **dwcinst** command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

Default values are stored in the `dwcinst.properties` file, located in the root directory of the installation image.

If you need to modify any of the default values, edit the `dwcinst.properties` file, but do not modify the `dwcinst.template` file located in the same path.

You can specify values in the properties file, type them in the command line, or use both methods. If a parameter is specified both in the properties file and in the command line, the command line value takes precedence.

Before starting the Dynamic Workload Console installation, ensure the following steps have been completed:

1. Installing WebSphere Application Server Liberty Base *(on page 37)* on the workstations where you plan to install the Dynamic Workload Console
2. Creating and populating the database *(on page 40)*
3. Creating the IBM Workload Scheduler administrative user *(on page 69)*
4. On UNIX™ operating systems, ensure that **umask** is set to **022**. To verify that **umask** is set to the correct value, from a command prompt, run the **umask** command. If the value is different from **022**, modify it by running the following command:

   ```
   umask 022
   ```

Note:

- Ensure that the **inst_dir** parameter is different from the directory of the installation image and it does not contain any IBM® Workload Scheduler instance.
- Recent JVMs do not fully support use of non-ASCII characters with the -jar and -javaagent commands. Use only ASCII characters in your installation directory names and paths.

To install the Dynamic Workload Console, perform the following steps:

1. Log in to the workstation where you plan to install the Dynamic Workload Console.
2. Download the installation images from IBM Fix Central.
3. Browse to the folder where the `dwcinst` command is located in *`image_location`*`/TWS/`*`interp_name`*.
4. Start the installation specifying a typical set of parameters:

   On Windows operating systems

   ```
   cscript dwcinst.vbs --acceptlicense yes --rdbmstype db_type
   --user dwc_admin_user --password dwc_pwd --dbname db_name
   --dbuser db_user --dbpassword db_pwd --dbhostname db_hostname
   --dbport db_port --wlpdir Liberty_installation_dir\wlp
   --sslkeysfolder certificate_files_path --sslpassword keystore_truststore_password
   ```

   On UNIX operating systems

   ```
   ./dwcinst.sh --acceptlicense yes --rdbmstype db_type
   --user dwc_admin_user --password dwc_pwd --dbname db_name
   --dbuser db_user --dbpassword db_pwd --dbhostname db_hostname
   --dbport db_port --wlpdir Liberty_installation_dir/wlp
   --sslkeysfolder certificate_files_path --sslpassword keystore_truststore_password
   ```

   where,

   user *dwc_admin_user*

   is the administrator of the Dynamic Workload Console. This user is added to the group of the Dynamic Workload Console administrators at installation time. You can use this account to log in to the Dynamic Workload Console and manage your environment.

   password *dwc_pwd*

   is the password of the Dynamic Workload Console user.

   On Windows operating systems

   Supported characters for the password are alphanumeric, dash (-), underscore (_) characters, and ()|?*~+.@!^

   On UNIX operating systems

   Supported characters for the password are any alphanumeric, dash (-), underscore (_) characters, and ()|?=*~+.

You have now successfully installed the Dynamic Workload Console.

Important: To ensure compatibility, the Dynamic Workload Console version installed must always be equal to or greater than the version of any engine it connects to.

For more information about all **dwcinst** parameters and default values, see Dynamic Workload Console installation - dwcinst script *(on page 320)*.

You can now proceed to Installing agents *(on page 79)*.

Installing agents

How to install an IBM Workload Scheduler fault-tolerant agent or dynamic agent in your distributed or end-to-end network by using the twsinst script.

When you install a fault-tolerant agent, also the remote command-line client is installed. You can use the client to run composer and conman commands.

Use only the twsinst script to install agents. If you are installing a dynamic agent, you can optionally add the Java™ run time which is needed to run job types with advanced options, and to configure a gateway to open communication with the dynamic workload broker.

When you install a dynamic or a fault-tolerant agent, also the following access methods, that extend the job scheduling capabilities of IBM Workload Scheduler to other software products, are installed:

PeopleSoft

To run and monitor PeopleSoft jobs from the IBM Workload Scheduler environment.

SAP

To create, schedule, and control SAP jobs by using the job scheduling features of IBM Workload Scheduler.

z/OS

To define and schedule jobs that run in a z/OS environment with JES2, JES3, or IBM Z Workload Scheduler

See Access methods *(on page)* for details about configuring and using the access methods.

Important: In order to be entitled to use the access methods and plug-ins, you must have purchased at least one of the following offerings: IBM Workload Scheduler, IBM Workload Scheduler for Applications, or IBM Z Workload Scheduler Agent. See the 10.2.3 Quick Start Guide available from IBM Fix Central. For information about the supported versions of the plug-ins and access methods, open the Data Integration report and select the **Supported Software** tab.

During each step of the installation process, the twsinst script creates files in the installation directory that you specified in the command. If you do not specify an installation directory in the `-inst_dir` option in the command, the script creates files in the following directories:

On Windows™ operating systems

`%ProgramFiles%\IBM\TWA_`*`TWS_USER`*

On UNIX™ operating systems

`/opt/IBM/TWA_`*`TWS_USER`*

Where *TWS_USER* is the user for which you are installing the IBM Workload Scheduler instance that you specify in the command.

The dynamic agent installation process automatically adds the workstation definition to the database and registers the workstation definition to the dynamic workload broker installed on the master domain manager or the dynamic domain manager that you specify during the installation process.

You can organize dynamic agents in pools to help organize your environment based on the availability of workstations and the requirements of the jobs to be run. Normally, when you create a pool, you add the dynamic agents to a workstation definition of type pool.

You can also register an agent with a pool by directly editing the `pools.properties` file located in `<`*`TWS_home`*`>/ITA/cpa/config`. See Automatically register agents to pools *(on page 145)* for more details.

To enable secure SSL communication for dynamic agents, you can choose one of the following methods:

- Download and deploy to dynamic agents the certificates already available on the master domain manager using the **wauser** and **wapassword** parameters when you run the twsinst installation script. Ensure the certificates are available on the master domain manager in the *TWA_DATA_DIR*`/ssl/depot` path.
- Use the **sslkeysfolder** and **sslpassword** parameters when you run the twsinst installation script. This applies to dynamic agents and fault-tolerant agents.

You only need to provide the path to the certificates and the password you want to define for the keystore and truststore. IBM® Workload Scheduler automatically generates the keystore and truststore with the specified password and configures WebSphere Application Server Liberty Base and your agents in SSL mode.

Enabling SSL during installation requires Java run time, which you can add at installation time using the **addjruntime** parameter, also available in the twsinst installation script. For more information, see Agent installation parameters - twsinst script *(on page 84)*.

At installation time, you can optionally create a subfolder on the master domain manager to contain one or more `*.crt` files to be added to the server truststore as trusted CA using the **sslkeysfolder** parameter. This can be used for example to add to the list of trusted CAs the certificate of the LDAP server or Db2 server. Additionally, you can store here any intermediate CA certificate to be added to the truststore. The subfolder must be named **additionalCAs**.

You can also use the **netmansslport** parameter when installing master domain manager, dynamic domain manager, and fault-tolerant agents to ensure **netman** communication between the server components and fault-tolerant agents takes place in SSL mode using the specified port number.

Agent installation procedure

1. Before you start to install, upgrade, or uninstall, verify that the user that runs the process has the following authorization requirements:

   Windows™ operating system

   If you set the Windows User Account Control (UAC), your login account must be a member of the Windows™ **Administrators** group or domain administrators with the rights **Act as Part of the Operating System**.

   If you set the Windows User Account Control (UAC) on the workstation, you must run the installation as **administrator**.

   UNIX™ and Linux™ operating systems

   You can choose to install agents as the **root** user, or as a **user other than root**. The following considerations apply:
   - If the installer is the **root user**, the **uname** parameter can be omitted if the *username* value is meant to be root, or can be set to a username value other than root.
   - If the installer is **different from the root user**, consider the following points:
     - The uname parameter can be omitted, but *username* is automatically set to the login name of the installer. If the installer specifies a uname with a different *username* value, an error message is returned.
     - As a consequence, the agent can run jobs uniquely with the user name of the installer.
     - The user must be enabled to login to the machine where the agent is going to be installed.
     - Event Management triggers on files work only if the selected files are accessible to the user that was used for the installation.
     - Future upgrades, modifications, and removal of the agent can be made exclusively with the same login used for installation. For dynamic agents, the login name used by the installer is stored in the read-only `InstallationLoginUser` parameter in the `JobManager.ini`configuration file on the agent.
     - When running **conman** and **composer** commands, it is mandatory to set the environment first, by using the tws_env script as described in Setting the environment variables *(on page 138)*.
2. Ensure that you downloaded the agent eImages (for details, see 10.2.3 Quick Start Guide available from IBM Fix Central.
3. Ensure that you have enough temporary space before you start the installation process.

You can install an agent in a distributed or an end-to-end environment.

To install an IBM Workload Scheduler agent, perform the following steps:

On Windows™ operating systems:

1. Download the agent eImage. For more information, see 10.2.3 Quick Start Guide available from IBM Fix Central.
2. Log in as administrator on the workstation where you want to install the product.
3. From the *image_directory*\TWS*operating_system* directory, run `twsinst` by using the following syntax:

   ```
   cscript twsinst.vbs -new -uname username -password user_password -acceptlicense yes
   ```

   For a description of the syntax parameters and a complete list of them, see Agent installation parameters - twsinst script *(on page 84)*.

   Note: twsinst for Windows™ is a Visual Basic Script (VBS) that you can run in CScript and WScript mode.

   The IBM Workload Scheduler user is automatically created. The software is installed by default in the IBM Workload Scheduler installation directory. The default value is `%ProgramFiles%\IBM\TWA`.

   If you enabled the Security Warning, a dialog box is displayed during the installation. In this case answer Run to continue.

On UNIX™ and Linux™ operating systems:

1. Download the agent eImage. For more information about eImages, see Downloading installation images on your workstation *(on page 157)* or the For more information, see 10.2.3 Quick Start Guide available from IBM Fix Central.
2. If you plan to login as **root** on the workstation where you will install the agent, create the IBM Workload Scheduler user. The software is installed by default in the user's home directory, referred to as */installation_dir/TWS*.

   User:

   TWS_user

   Home:

   /installation_dir/TWS (for example: `/home/user1/TWS` where `user1` is the name of IBM Workload Scheduler user). Ensure this directory has **755** permission.

   If you plan to log in as a **non-root user**, your login will become by default the only possible user of the agent. You do not need to create another IBM Workload Scheduler user, but make sure that you have a home directory (where the agent will be installed), and that it has **755** permission.

   Important: If you use the `-su non-root username` command in the shell where you are about to run `twsinst`, make sure that $HOME is set on your home directory as a non-root user (use `echo $HOME` to verify that the value returned corresponds to your home directory).

3. Log in on the workstation where you want to install the product.
4. From the *image_directory*/TWS/*operating_system* directory, run **twsinst** by using the following syntax:

   ```
   ./twsinst -new -uname username -acceptlicense yes
   ```

   For a description of the syntax parameters, see Agent installation parameters - twsinst script *(on page 84)*

If the installation fails, to understand the cause of the error see Analyzing return codes for agent installation, upgrade, restore, and uninstallation *(on page 280)*.

After a successful installation, perform one of the following configuration tasks, depending on the type of agent you installed:

- Configuring a fault-tolerant agent *(on page 98)*.
- Configuring a dynamic agent *(on page 145)*.

On Windows™ operating systems:

Show command usage and version

```
cscript twsinst.vbs -u | -v
```

Install a new instance

```
cscript twsinst.vbs -new
   -acceptlicense yes|no
   -uname  username
   [-domain user_domain]
    -password user_password

   [-agent dynamic|fta|both]

   [-addjruntime true|false]
   [-inst_dir install_dir]
   [-lang lang_id]
   [-skipcheckprereq]
   [-skip_usercheck]
   [-work_dir working_dir]

   [-agentid id]
   [-company company_name]
   [-master master_cpu_name]
   [-port port_number]
   [-netmansslport port_number]
   [-thiscpu workstation]
   [-encryptionpassword password]
   [-useencryption boolean]

   [-gateway local|remote|none]
   [-gwid gateway_id]
   [-gweifport gateway_eif_port]

   [-displayname agentname]
   [-hostname host_name]
   [-jmport port_number]
   [-jmportssl true|false]
   -tdwbhostname host_name
   -tdwbport tdwbport_number

   [-sslpassword ssl_password]
   [-sslkeysfolder ssl_folder]

   [-jwt true | false]
   [-wauser wauser -wapassword wapassword] | [-apikey apikey]
```

On UNIX™ and Linux™ operating systems

Show command usage and version

```
./twsinst -u | -v
```

Install a new instance

```
./twsinst -new
   -acceptlicense yes|no
   [-reset_perm]
   [-uname username]
   [-data_dir data_directory]
   [-agent dynamic|fta|both]
   [-addjruntime true|false]
   [-inst_dir install_dir]
   [-lang lang_id]
   [-skipcheckprereq]
   [-skip_usercheck]
   [-work_dir working_dir]
   [-agentid id]
   [-company company_name]
```

```
[-master master_cpu_name]
[-port port_number]
[-netmansslport port_number]
[-thiscpu workstation]
[-encryptionpassword password]
[-useencryption boolean]
[-gateway local|remote|none]
[-gwid gateway_id]
[-gweifport gateway_eif_port]
[-displayname agentname]
[-hostname host_name]
[-jmport port_number]
[-jmportssl true|false]
-tdwbhostname host_name
-tdwbport tdwbport_number
[-create_link]
[-sslpassword ssl_password]
[-sslkeysfolder ssl_folder]
[-jwt true | false]
[-wauser wauser -wapassword wapassword] | [-apikey apikey]
```

Agent installation parameters - twsinst script

Agent installation parameters that can be passed to the twsinst script.

This section lists and describes the parameters that are used when running a twsinst script to install dynamic agents, fault-tolerant agents.

Certificates are now required when installing or upgrading IBM® Workload Scheduler. You can no longer install nor upgrade IBM® Workload Scheduler without securing your environment with certificates. The required certificates are:

- ca.crt
- tls.key
- tls.crt

For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).

To find some sample agent installation scenarios, see Example installation commands *(on page 92)* and Dynamic agent gateway installation examples *(on page 94)*.

-acceptlicense *yes*|*no*

Specify whether to accept the License Agreement.

-addjruntime *true*|*false*

Adds the Java™ run time to run job types with advanced options, both those types that are supplied with the product and the additional types that are implemented through the custom plug-ins. Valid values are **true** and **false**. The default for a fresh installation is **true**. Set this parameter to `true` if you use the **sslkeysfolder** and **sslpassword** parameters to define custom certificates in PEM format.

If you decided not to install Java™ run time at installation time, you can still add this feature later as it is described in Adding a feature *(on page 152)*.

-agent *dynamic*|*fta*|*both*

The type of agent that you want to install. Valid values are:

dynamic

To install the IBM Workload Scheduler dynamic agent. Use this value with the **-tdwbhostname** *host_name* and the **-tdwbport** *tdwbport_number* parameters.

On Windows operating systems, you can install dynamic agents using the Local System Account. To install with the Local System Account, omit the **uname** and **password** parameters.

fta

To install the IBM Workload Scheduler fault-tolerant agent.

both

To install the dynamic agent that is used with the **-tdwbhostname** *host_name* and the **-tdwbport** *tdwbport_number* parameters, and a fault-tolerant agent.

The default is ***dynamic***.

-agentid ***agent_id***

The unique identifier of the agent that you want to install. The parameter is optional. If not specified, the installation process assigns to the agent a string of alphanumeric characters, as in the following example:

```
893164748CCA4FC6820F12685AECBB07
```

It might be useful to specify an *agent_id* when you want to reinstall an agent after it was uninstalled, and you want to use the same *agent_id*. This prevents that two different *agent_id* values are registered on the server for the same agent installation.

Note: When you manually specify the agent_id value, ensure that the length is 32 characters, otherwise an error occurs.

If you set the **jwt** parameter to `true`, the **agentId** parameter is ignored if provided, because the agent ID is retrieved from the master domain manager together with the JWT. See -jwt true | false *(on page 87)*.

-apikey

Use this parameter to specify the API key to be used for authenticating with the master domain manager. This authentication enables downloading the certificates or JWT to be used for communication between dynamic agent and dynamic domain manager. This parameter is mutually exclusive with the **wauser** and **wapassword** parameters. A random password in base64 encoding is automatically created for generating stash files. The password stored in the `tls.sth` file. If needed, you can decrypt this password using any base64 decoder.

Obtain the string to be provided with this parameter from the Dynamic Workload Console before running the command. For more information, see Authenticating the command line client using API Keys *(on page)*.

-company ***company_name***

The name of the company. The company name cannot contain blank characters. The name is shown in program headers and reports. If not specified, the default name is COMPANY.

-create_link

UNIX™ systems only. Create the **symlink** between `/usr/bin/at` and *`install_dir`*`/TWS/bin/at`. For more information, see Table 2: Symbolic link options *(on page 22)*.

-data_dir ***path***

This argument applies to UNIX operating systems only. Specify a path for product data, such as log and configuration files, if you want to install the product binaries separated from the product data. This argument is optional. The default value is *`INSTALL_DIR`*`/TWSDATA`.

-displayname ***display_name***

The name to assign to the agent. The name cannot start with a number. The default is based on the host name of this computer.

If the host name starts with a number, the **-displayname** parameter must be specified.

-domain ***user_domain***

Windows™ systems only. The domain name of the IBM Workload Scheduler user. The default is the name of the workstation on which you are installing the product. Ensure you use `USERDOMAIN` instead of `USERDNSDOMAIN`.

-enablefips ***true|false***

Specify whether you want to enable FIPS. In the current product version, you can only specify `false` because FIPS is not supported. In a fresh installation, the default is `false`. In upgrade, there is no default value, so you

have to set it explicitly and be aware that FIPS is being disabled when you upgrade. This parameter is optional. If you are upgrading from an environment where FIPS is supported, see Q: My environment is FIPS compliant. What happens if I upgrade to version 10.2.3? *(on page 270)*.

-encryptionpassword ***default***

The password for the keystore storing the AES-256 or AES-128 keys used to encrypt the files at runtime. This parameter is optional. The default value is *default*. You can optionally encrypt the password using the secure script. For more information, see Optional password encryption - secure script *(on page 300)*.

-gateway ***local*|*remote*|*none***

Specifies whether to configure a gateway to communicate with the dynamic workload broker or not, and how it is configured. Specify *local* if the gateway is local to the dynamic agent workstation. Specify *remote* if the dynamic agent communicates through a gateway that is installed on a different dynamic agent workstation from the dynamic agent being installed. The default value is *none*, which means no gateway is configured. For information about installing with a local and remote gateway, see Example installation commands *(on page 92)*.

-gweifport ***gateway_eif_port***

Specifies the Job Manager Event Integration Facility (EIF) port number. The default value is **31132**. The valid range is 1 to 65535.

-gwid ***gateway_id***

The unique identifier for the gateway. This parameter is required when you specify **-gateway** *local*. The default gateway identifier that is assigned is **GW1**. The gateway identifier must start with either an alphabetic character or an underscore character (_), and it can contain only the following types of characters: alphabetic, numeric, underscores (_), hyphens (-), and periods (.).

Gateways can also work in parallel to mutually take over in routing communications to the agents connected to them. To enable gateways to work in parallel, all gateways must have the same *gateway_id* assigned. This information is stored in the `JobManagerGW.ini` file, by setting the JobManagerGWURIs property.

-hostname ***host_name***

The fully qualified hostname or IP address on which the agent is contacted by the dynamic workload broker. The default is the hostname of this computer. If the hostname is a localhost, the hostname parameter must be specified.

-inst_dir ***installation_dir***

The directory of the IBM Workload Scheduler installation.

On Windows™ operating systems:

If you specify a path that contains blanks, enclose it in double quotation marks. Specify an absolute path. If you do not manually specify a path, the path is set to `%ProgramFiles%\IBM\TWA_`*`TWS_USER`*, where *TWS_USER* is the user for which you are installing the IBM Workload Scheduler that you specify in the **-uname** parameter. If you use the Local System Account and therefore do not specify the **-uname** parameter, the path is set to `%ProgramFiles%\IBM\TWA_WaLocalSystemAccount`.

On UNIX™ and Linux™ operating systems:

If you specify a path that contains blanks, enclose it in double quotation marks. Specify an absolute path. If you do not manually specify a path, the path is set to:

- `/opt/IBM/TWA_`*`TWS_USER`*, if you logged in as the **root** user to install the agent. *TWS_USER* is the user that you specify in the `-uname` option and for which you are installing the agent (can omit if *TWS_USER* is **root**). The IBM Workload Scheduler user that you specify in the `-uname username` parameter must have read and run privileges for the `installation_dir` installation path; otherwise the installation fails.
- *`home_dir`*`/TWA`, if you logged in with a login **other than root**. Ensure that the directory permission is set to **755** for *home_dir*, the home directory for your login, and that you are the *home_dir* owner.

-jmport ***port_number***

The JobManager port number used by the dynamic workload broker to connect to the dynamic agent. The default value is **31114**. The valid range is from 1 to 65535.

-jmportssl ***true|false***

The JobManager port used by the dynamic workload broker to connect to the IBM Workload Scheduler dynamic agent. The port value is the value of the ssl_port parameter in the `ita.ini` file if **-jmportssl** is set to `true`. If set to `false`, it corresponds to the value of the **tcp_port** parameter in the `ita.ini` file. The `ita.ini` file is located in `ITA\cpa\ita` on Windows™ systems and `ITA/cpa/ita` on UNIX™, Linux™, and IBM i systems.

Set the value to "true" if **- gateway** is set to `local`.

For communication using SSL or HTTPS

Set **jmportssl = *true***. To communicate with the dynamic workload broker, it is recommended that you set the value to *true*. In this case, the port specified in **jmport** communicates in HTTPS.

For communication without using SSL or through HTTP

Set **jmportssl = *false***. In this case the port specified in **jmport** communicates in HTTP.

-jwt *true* | *false*

Specify `true` to use the JSON Web Token (JWT) to authenticate with the master domain manager. Specify `false` to authenticate with the master domain manager using certificates. The default value is `true`. This parameter is mutually exclusive with the **sslkeysfolder** and **sslpassword** parameters which are used to generate custom certificates.

If you set this parameter to `true`, the **agentId** parameter is ignored if provided, because the agent ID is retrieved from the master domain manager together with the JWT. See -agentid agent_id *(on page 85)*. Also, if you set this parameter to `true`, the following parameters are required for downloading the JWT:

- **wauser** or **apikey**. See -wauser wauser_name *(on page 91)* or -apikey *(on page 85)*.
- **wapassword** or **apikey**. See -wapassword wauser_password *(on page 91)* or -apikey *(on page 85)*.
- **tdwbhostname**. See -tdwbhostname host_name *(on page 89)*.
- **tdwbport**. See -tdwbport tdwbport_number *(on page 89)*.

For examples of installations with JWT, see Example installation commands *(on page 92)*.

-lang *lang_id*

The language in which the twsinst messages are displayed. If not specified, the system LANG is used. If the related catalog is missing, the default C language catalog is used. If neither **-lang** nor LANG are used, the default codepage is set to SBCS. For a list of valid values for these variables, see the following table:

Table 6. Valid values for -lang and LANG parameter

Language	Value
Brazilian portuguese	pt_BR
Chinese (traditional and simplified)	zh_CN, zh_TW
English	en
French	fr
German	de
Italian	it
Japanese	ja
Korean	ko
Russian	ru
Spanish	es

Note: This is the language in which the installation log is recorded and not the language of the installed engine instance. twsinst installs all languages as default.

-master ***workstation***

The workstation name of the master domain manager. This name cannot exceed 16 characters, cannot contain spaces, and cannot be the same as the workstation name that you entered in the **thiscpu** parameter. If not specified, the default value is **MASTER**.

-new

A fresh installation of the agent. Installs an agent and all supported language packs.

-password ***user_password***

Windows™ systems only. The password of the user for which you are installing IBM Workload Scheduler. The password can include alphanumeric, dash (-), and underscore (_) characters, and the following symbols: ()!? =^*/~ [] $`+;:.@. The **-password** parameter is used for fresh installations only, it is not required for fix packs or upgrades. You can optionally encrypt the password using the secure script. For more information, see Optional password encryption - secure script *(on page 300)*.

On Windows operating systems, you can install dynamic agents using the Local System Account. To install with the Local System Account, omit the **uname** and **password** parameters.

-netmansslport ***SSL_port_number***

The TCP/IP port number used by the netman process to listen for communication from the master in SSL mode. The default value is 31113. The valid range is from 1 to 65535. You can also set the **netmansslport** parameter to `disabled` to use non-encrypted communication. If you set the **netmansslport** parameter to `disabled`, you must provide a value for the **netmanport** parameter. This port number is registered in the `localopts` file, in the **nm ssl full port** attribute. For each installation you must specify a different number.

-port ***port_number***

The TCP/IP port number used by the Netman process to listen for communication from the master. The default value is **31111**. The valid range is from 1 to 65535. This port number is registered in the `localopts` file. For each installation you must specify a different number.You can also set this parameter to `disabled`. In this case, you must provide a value for the **netmansslport** parameter, which enables SSL communication.

-reset_perm

UNIX™ and IBM i systems only. Reset the permission of the libraries in the `/usr/ibm` directory.

-restore

Run this command from the folder to where you copied the eImage (a folder other than the home directory of *TWS_USER*, where *TWS_USER* is the user that installed the IBM Workload Scheduler instance), and not from the installation path, to restore the version in the eImage.

-skip_usercheck

Enable this option if the authentication process within your organization is not standard, thereby disabling the default authentication option.

On Windows™ systems, if you specify this parameter, the program does not create the user you specified in the **-uname** *username* parameter and you must create the user manually before running the script. However, if you use Local System Account, you do not need to specify any user.

On UNIX™ and Linux™ systems if you specify this parameter, the program skips the check of the user in the `/etc/passwd` file or the check you perform using the su command.

-skipcheckprereq

If you specify this parameter, IBM Workload Scheduler does not scan system prerequisites before installing the agent. For more information on the prerequisite check, see Scanning system prerequisites for IBM Workload Scheduler *(on page 33)*.

-sslkeysfolder ***path***

The name and path of the folder on the agent containing PEM certificates. The installation program automatically generates the keystore and truststore files using the password you specify with the **--sslpassword** parameter.

The folder must contain the following files and folders:

ca.crt

The Certificate Authority (CA) public certificate.

tls.key

The private key for the instance to be installed.

tls.crt

The public part of the previous key.

tls.sth

The file storing your encoded password in Base64 encoding.

You can optionally create a subfolder to contain one or more `*.crt` files to be added to the server truststore as trusted CA. This can be used for example to add to the list of trusted CAs the certificate of the LDAP server or DB2 server. Additionally, you can store here any intermediate CA certificate to be added to the truststore. The subfolder must be named **additionalCAs**. If you are connecting a master domain manager using custom certificates to a dynamic agent also using custom certificates, the only required file is `ca.crt`.

Before you start the installation, ensure the required files and folders are available on the agent.

The **sslkeysfolder** and **sslpassword** parameters are mutually exclusive with the **wauser**, **wapassword**, and **jwt** parameters, which are used to download and deploy the certificates or JWT already available on the master domain manager.

-sslpassword ***password***

Specify the password for the certificates in PEM format automatically generated by the installation program.

If you use this parameter, ensure that the **addjruntime** parameter is set to true, because Java™ run time is required for defining custom certificates.

You can optionally encrypt the password using the secure script. For more information, see Optional password encryption - secure script *(on page 300)*.

-tdwbhostname ***host_name***

The fully qualified host name or IP address of the dynamic workload broker the agent is registering to. It applies when you set the **-agent** parameter to either **dynamic** or **both** and requires the **-tdwbport** parameter. This parameter is required.

If you set the **-gateway** parameter to `remote`, this is the host name of the dynamic agent hosting the gateway and to which the agent you are installing will connect. This information is stored in the `JobManager.ini` file. For information about installing with a local and remote gateway, see Example installation commands *(on page 92)*.

If you set the **jwt** parameter to `true`, ensure you provide a value for this parameter, so that you can download the JWT and agent ID from the specified dynamic domain manager. The dynamic domain manager routes the JWT request to the master domain manager. See also -jwt true | false *(on page 87)*.

-tdwbport ***tdwbport_number***

The HTTPS transport port number of the dynamic workload broker the agent is registering to. It must match the port you specified with **httpsport** parameter when installing the master domain manager. It applies when you set the **-agent** parameter to either **dynamic** or **both** and requires the **-tdwbhostname** parameter. The valid range is from 0 to 65535. If you specify `0` you cannot run workload dynamically. Do not specify `0` if the *-agent* value is `dynamic` or `both`. The default is `0` for an upgrade, which means that this connection is not configured, otherwise, specify `31116` for a fresh installation. This parameter is required.

If you set the **-gateway** parameter to `remote`, this is the HTTP or HTTPS port number of the dynamic agent hosting the gateway and to which the agent you are installing will connect. You have specified this port with the

jmport parameter when installing the agent hosting the gateway. For information about installing with a local and remote gateway, see Example installation commands *(on page 92)*.

If you are performing a fresh installation, the value to use is `31114`. This information is stored in the `JobManager.ini` file.

If you set the **jwt** parameter to `true`, ensure you provide a value for this parameter, so that you can download the JWT and agent ID from the specified dynamic domain manager. The dynamic domain manager routes the JWT request to the master domain manager. See also -jwt true | false *(on page 87)*.

-thiscpu *workstation*

The name of the IBM Workload Scheduler workstation of this installation. The name cannot exceed 16 characters, cannot start with a number, cannot contain spaces, and cannot be the same as the workstation name of the master domain manager. This name is registered in the `localopts` file. If not specified, the default value is the host name of the workstation.

If the host name starts with a number, **-thiscpu** parameter must be specified.

-u

Displays command usage information and exits.

-uname *username*

The name of the user for which the IBM Workload Scheduler agent is being installed. This user owns the IBM Workload Scheduler instance and by default, jobs are run with its name. This user name is not to be confused with the user performing the installation, unless you use a **user other than root**. The user name cannot contain periods (.).

On UNIX™ and Linux™ systems, for a new installation, this user account must be created manually before running the installation and must be enabled to login to the machine where the agent is going to be installed. Create a user with a home directory. IBM Workload Scheduler is installed by default under the home directory of the specified user.

You can choose to install agents as the **root** user, or as a **user other than root**. The following considerations apply:

- If the installer is the **root user**, the **uname** parameter can be omitted if the *username* value is meant to be root, or can be set to a username value other than root.
- If the installer is **different from the root user**, consider the following points:
  - The uname parameter can be omitted, but *username* is automatically set to the login name of the installer. If the installer specifies a uname with a different *username* value, an error message is returned.
  - As a consequence, the agent can run jobs uniquely with the user name of the installer.
  - The user must be enabled to login to the machine where the agent is going to be installed.
  - Event Management triggers on files work only if the selected files are accessible to the user that was used for the installation.
  - Future upgrades, modifications, and removal of the agent can be made exclusively with the same login used for installation. For dynamic agents, the login name used by the installer is stored in the read-only `InstallationLoginUser` parameter in the `JobManager.ini`configuration file on the agent.
  - When running **conman** and **composer** commands, it is mandatory to set the environment first, by using the tws_env script as described in Setting the environment variables *(on page 138)*.

On Windows operating systems, you can install dynamic agents using the Local System Account. To install with the Local System Account, omit the **uname** and **password** parameters.

-useencryption true | false

Specifies whether IBM® Workload Scheduler files must be encrypted at runtime. If you specify *true*, or do not set this parameter, files such as the Symphony file and the message queues are encrypted using AES-256 or AES-128 cryptography. By default, a fresh installation is automatically encrypted and the keystore password is *default*. To change the keystore password, use the **encryptionpassword** parameter. This parameter is optional.

-wauser *wauser_name*

One of the following users, defined on the master domain manager:

- The user for which you have installed the master domain manager the agent is connecting to.
- The user with the DISPLAY permission on the FILE named AGENT_CERTIFICATE. This permission allows the user to download certificates or JWT. For more information about this scenario, see Downloading certificates or JWT using a different user *(on page 342)*.

Always specify the user defined on the master domain manager, also if you are installing a dynamic agent and want it to register to a dynamic domain manager. This is because the dynamic domain manager simply forwards data to and from the master domain manager.

By providing the **wauser** and **wapassword** parameters or the **apikey** parameter, you enable IBM Workload Scheduler to download and install either the certificates in PEM format or the JWT already available on the master domain manager. To download PEM certificates, set the **jwt** parameter to `false`, to download JWT, set the **jwt** parameter to `true`. For more information, see -jwt true | false *(on page 87)*.

This parameter is mutually exclusive with the **apikey** parameter, which provides authentication using an API Key.

For more information, see -apikey *(on page 85)*.

This parameter is also mutually exclusive with the **sslkeysfolder** parameter, which is used to specify a folder on the agent where you store the certificates. For more information, see -sslkeysfolder path *(on page 88)*.

For further information about how to automatically download and deploy certificates in PEM format from the master domain manager to dynamic agents and fault-tolerant agents, see Certificates download to dynamic agents and fault-tolerant agents - AgentCertificateDownloader script *(on page 338)*.

-wapassword *wauser_password*

One of the following passwords, defined on the master domain manager:

- The password of the user for which you have installed the master domain manager the agent is connecting to.
- The password of the user with the DISPLAY permission on the FILE named AGENT_CERTIFICATE. This permission allows the user to download certificates or JWT. For more information about this scenario, see Downloading certificates or JWT using a different user *(on page 342)*.

Always specify the user defined on the master domain manager, also if you are installing a dynamic agent and want it to register to a dynamic domain manager. This is because the dynamic domain manager simply forwards data to and from the master domain manager.

By providing the **wauser** and **wapassword** parameters or the **apikey** parameter, you enable IBM Workload Scheduler to download and install either the certificates in PEM format or the JWT already available on the master domain manager. To download PEM certificates, set the **jwt** parameter to `false`, to download JWT, set the **jwt** parameter to `true`.

See also -jwt true | false *(on page 87)*.

This parameter is mutually exclusive with the **apikey** parameter, which provides authentication using an API Key.

For more information, see -apikey *(on page 85)*.

This parameter is also mutually exclusive with the **sslkeysfolder** parameter, which is used to specify a folder on the agent where you store the certificates. You can optionally encrypt the password using the secure script. For more information, see Optional password encryption - secure script *(on page 300)*.

For further information about how to automatically download and deploy certificates in PEM format from the master domain manager to dynamic agents and fault-tolerant agents, see Certificates download to dynamic agents and fault-tolerant agents - AgentCertificateDownloader script *(on page 338)*.

-work_dir ***working_dir***

The temporary directory used by the program to deploy the installation process files.

On Windows™ operating systems:

If you specify a path that contains blanks, enclose it in double quotation marks. If you do not manually specify a path, the path is set to `%temp%\TWA\tws`*`version_number`*, where `%temp%` is the temporary directory of the operating system.

On UNIX™ and Linux™ operating systems:

The path cannot contain blanks. If you do not manually specify a path, the path is set to `/tmp/TWA/tws`*`version_number`*.

This parameter can also function as a backup directory during product upgrade with path *`WORKING_DIR`*`/backup` if you do not set the **-skipbackup** parameter to **true**.

-v

Displays the command version and exits.

Example installation commands

Consider the following examples to understand the use and capabilities of the twsinst command. You can optionally encrypt the password using the secure script. For more information, see Optional password encryption - secure script *(on page 300)*.

- The following example shows the syntax used when using the **twsinst** script to install a new instance of a dynamic agent and adding the Java™ run time for running job types with advanced options.

  On Windows™ operating systems:

  ```
  cscript twsinst.vbs -new
     -uname TWSuser1 -password user_password -acceptlicense yes
     -addjruntime true -agent dynamic -displayname thishostcomputername
     -hostname thishostname.mycompany.com -wauser wauser -wapassword wapassword
     -inst_dir "c:\Program Files\IBM\TWA_TWSuser1"
     -jmport 31114 -tdwbport 31116 -tdwbhostname mainbroker.mycompany.com
  ```

  On UNIX and Linux™ operating systems:

  ```
  ./twsinst -new
     -uname TWSuser1 -acceptlicense yes -addjruntime true
     -agent dynamic -displayname thishostcomputername
     -hostname thishostname.mycompany.com -wauser wauser -wapassword wapassword
     -inst_dir "/opt/IBM/TWA_TWSuser1"
     -jmport 31114 -reset_perm -skipcheckprereq -tdwbport 31116
     -tdwbhostname mainbroker.mycompany.com
  ```

- The following example shows the syntax used when running the **twsinst** script to install a new instance of both a fault-tolerant and a dynamic agent, and adding the Java™ run time for running job types with advanced options. Ensure you copy the certificates on the agent before you start the installation. The path to the certificates is specified with the **sslkeysfolder** parameter. The **sslpassword** parameter specifies the password to access the certificates. In this case, the **jwt** parameter must be set to `false`:

  On Windows™ operating systems:

  ```
  cscript twsinst.vbs -new
     -uname TWSuser1 -password user_password -acceptlicense yes
     -addjruntime true -agent both -displayname thishostcomputername
     -hostname thishostname.mycompany.com -sslkeysfolder /MyCertsFolder -jwt=false
     -sslpassword fer1smx24569ijDCS86?! -inst_dir "c:\Program Files\IBM\TWA_TWSuser1"
     -jmport 31114 -master TWSmdm -tdwbport 31116 -tdwbhostname mainbroker.mycompany.com
     -thiscpu mainworkstation
  ```

  On UNIX™ and Linux™ operating systems:

  ```
  ./twsinst -new
     -uname TWSuser1 -acceptlicense yes -addjruntime true
     -agent both -create_link -displayname thishostcomputername
  ```

```
   -hostname thishostname.mycompany.com -inst_dir "/opt/IBM/TWA_TWSuser1"
   -sslkeysfolder /MyCertsFolder -sslpassword fer1smx24569ijDCS86?!
   -jmport 31114 -master TWSmdm -reset_perm -skipcheckprereq
   -tdwbport 31116 -tdwbhostname mainbroker.mycompany.com
   -thiscpu fta101
```

- The following example shows the syntax used when using the **twsinst** script to install a new instance of a dynamic agent, adding the Java™ run time for running job types with advanced options, and to install a gateway on the same workstation as the agent to enable communication with the master domain manager.

  On Windows™ operating systems:

```
cscript twsinst.vbs -new
   -uname TWSuser1 -password user_password -acceptlicense yes
   -addjruntime true -agent dynamic -displayname thishostcomputername
   -gateway local -gwid gateway_id
   -hostname thishostname.mycompany.com
   -inst_dir "c:\Program Files\IBM\TWA_TWSuser1"
   -wauser MDMAdmin -wapassword -wapassword 547832gtrOLK8542Mnfdw!
   -jmport 31114 -jimportssl true -master TWSmdm -skipcheckprereq
   -tdwbport 31116 -tdwbhostname mainbroker.mycompany.com
   -thiscpu mainworkstation
```

  On UNIX™ and Linux™ operating systems:

```
./twsinst -new
   -uname TWSuser1 -acceptlicense yes -addjruntime true -agent both
   -displayname thishostcomputername -create_link -gateway local
   -gwid gateway_id
   -hostname thishostname.mycompany.com -inst_dir "/opt/IBM/TWA_TWSuser1"
   -wauser MDMAdmin -wapassword -wapassword 547832gtrOLK8542Mnfdw!
   -jmport 31114 -jimportssl true -master TWSmdm -reset_perm -skipcheckprereq
   -tdwbport 31116 -tdwbhostname mainbroker.mycompany.com
   -thiscpu fta101
```

- In the following example, you install a new agent with both dynamic agent and fault-tolerant agent capabilities.

  By setting **jwt** to `true`, you install the agent and authenticate with the master domain manager using JWT. The **jwt** parameter requires the **tdwbhostname** and **tdbwport** parameters for connecting to the dynamic domain manager and the **wauser** and **wapassword** parameters, which provide the credentials to be used when logging in to the master domain manager. The credentials are required when you first download the JWT from the master domain manager:

```
twsinst -new -acceptlicense yes -agent both -uname TWSuser1 -tdwbhostname Saturn
-tdwbport 37116 -master Jupiter -jwt true -wauser MDMAdmin -wapassword 125784gtrOLK8542Mnfdw!
```

- This example is a variation from the previous example. Instead of using the **wauser** and **wapassword** parameters when logging in to the master domain manager, you authenticate using the API Key you have previously retrieved from the Dynamic Workload Console. For this purpose, use the **apikey** parameter:

```
twsinst -new -acceptlicense yes -agent both -uname TWSuser1 -tdwbhostname Saturn
-tdwbport 37116 -master Jupiter -jwt true -apikey eyJraWQiOiJha2li....KINijWmdC-fY
```

- In the following example, you install a dynamic agent and authenticate with the master domain manager named `Saturn` using JWT. The agent hosts a **local** gateway with gateway ID `GWD1`:

```
twsinst -new -acceptlicense yes -agent dynamic -uname TWSuser1 -hostname Titan -displayname Titan
-tdwbhostname Saturn -tdwbport 37116 -jwt true -wauser MDMAdmin
-wapassword 125784gtrOLK8542Mnfdw! -gateway local -gwid GWID1 -jmport 42427
```

- In the following example, you install an agent and authenticate with the master domain manager using JWT. The agent connects to the **remote** gateway installed on the agent in the previous example. In this case, the **tdwbhostname** parameter must be set to the host name of the dynamic agent (`Titan`) hosting the gateway and to which the agent you are installing will connect. In the same way, the **tdwbport** parameter must match the port number of the dynamic agent hosting the gateway. You have set this port using the **jmport** parameter when installing the dynamic agent hosting the gateway.

```
twsinst -new -acceptlicense yes -agent both -uname TWSuser1 -displayname Tetis
-tdwbhostname Titan -tdwbport 42427 -jwt true -wauser MDMAdmin
-wapassword 125784gtrOLK8542Mnfdw! -gateway remote -jmport 54548
```

- In the following example, you install the agent using local certificates for authentication. Ensure you copy the certificates on the agent before you start the installation. The path to the certificates is specified with the **sslkeysfolder** parameter. The **sslpassword** parameter specifies the password to access the certificates. In this case, the **jwt** parameter must be set to `false`:

```
twsinst -new -acceptlicense yes -agent both -uname TWSuser1 -tdwbhostname Saturn
-tdwbport 37116 -master Jupiter -jwt false -sslkeysfolder /MyCertsFolder
-sslpassword fer1smx8854ijDSW65?!
```

- In the following example, you install the agent and download to the agent the certificates from the master domain manager. Ensure the certificates are available on the master domain manager in one of the following paths:

  On Windows operating systems

  ```
  installation_directory\TWS\ssl\depot
  ```

  On UNIX operating systems

  ```
  TWA_DATA_DIR/ssl/depot
  ```

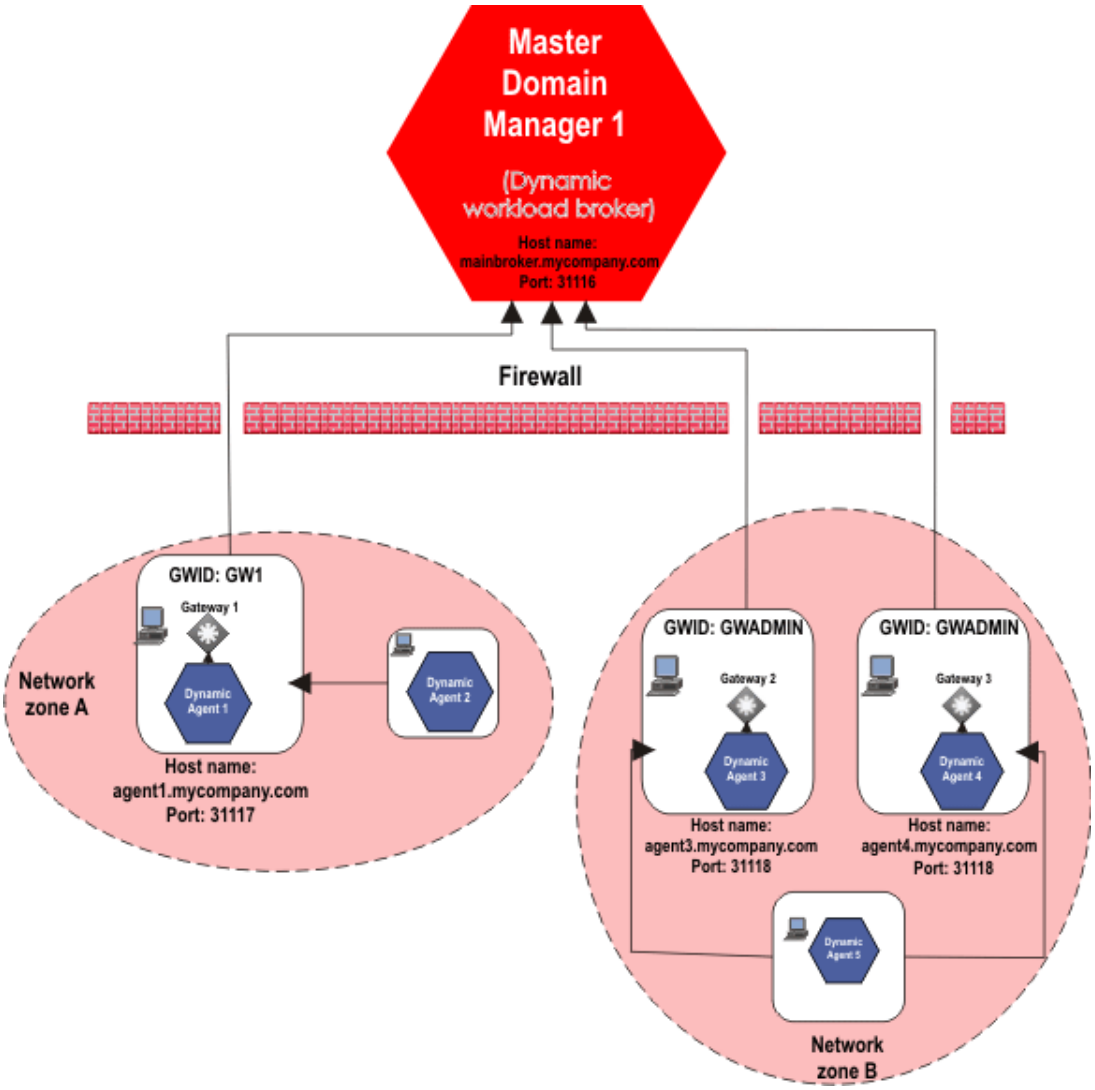
  To download the certificates, specify the **wauser** and **wapassword** parameters to connect to the master domain manager. In this case, the **jwt** parameter must be set to `false`

```
twsinst -new -acceptlicense yes -agent both -uname TWSuser1 -tdwbhostname Saturn -tdwbport 37116
-master Jupiter -jmport 1234 -port 1235 -netmansslport 1236 -jwt false -wauser MDMAdmin
-wapassword 125784gtrOLK8542Mnfdw!
```

Dynamic agent gateway installation examples

Example installations for configuring a local or remote gateway with dynamic agent workstations in the same or different network zones.

The following examples address two installation scenarios and indicate the parameters to use with the twsinst script to install the dynamic agents to support the scenarios. The following figure depicts the two scenario environments:



Scenario 1: Same network zone

The workstations where you install the agents can communicate with each other (Dynamic Agent 1 and Dynamic Agent 2) and are located in the same network zone, but only one agent workstation (Dynamic agent 1) can connect to the dynamic workload broker.

Table 7. Installation syntax for agent installation with agents in the same network zone

Dynamic Agent workstation	Installation syntax

Dynamic Agent 1	`twsinst -new -uname <user_name>`
	`  -password <user_password>`
	`  -acceptlicense yes`
	`  -agent dynamic`
	`  -gateway local`
	`  -gwid GW1`
	`  -jmport 31117`
	`  -tdwbport 31116`
	`  -tdwbhostname mainbroker.mycompany.com`
	`  -wauser wauser`
	`  -wapassword password`

Table 7. Installation syntax for agent installation with agents in the same network zone (continued)

Dynamic Agent workstation	Installation syntax

Dynamic Agent 2	`twsinst -new -uname <user_name>`
	`  -password user_password>`
	`  -acceptlicense yes`
	`  -agent dynamic`
	`  -gateway remote`
	`  -tdwbport 31117`
	`  -tdwbhostname agent1.mycompany.com`

where,

Dynamic Agent 1

-gateway local

Dynamic Agent 1 communicates with the dynamic workload broker through its local gateway.

-gwid GW1

The gateway ID is the name that identifies the gateway site on Dynamic Agent 1. The default name is GW1.

-tdwbport 31116

The port number of the dynamic workload broker.

-tdwbhostname mainbroker.mycompany.com

The fully qualified host name of the dynamic workload broker.

Dynamic Agent 2

-gateway remote

Indicates that Dynamic Agent 2 can connect to the internet through a gateway installed on a different agent, Dynamic Agent 1.

-tdwbport 31117

The port number of the dynamic agent workstation where the gateway resides. In this example, the port number of Dynamic Agent 1 is 31117.

-tdwbhostname agent1.mycompany.com

The fully qualified host name of the dynamic agent workstation where the gateway resides and to which the agent connects.

Scenario 2: Different network zones

The workstations where you install the agents cannot communicate with each other and are in different network zones (Network zone A and Network zone B), however, one agent workstation in each network zone can successfully connect to the dynamic workload broker. In Network zone B, two parallel gateways are configured.

Table 8. Installation syntax for agent installation with agents in different network zones

Dynamic Agent workstation	Installation syntax

Dynamic Agent 3	`twsinst -new -uname <user_name>`
	`  -password user_password`
	`  -acceptlicense yes`
	`  -agent dynamic`
	`  -gateway local`
	`  -gwid GWADMIN`
	`  -jmport 31118`
	`  -tdwbport 31116`
	`  -tdwbhostname mainbroker.mycompany.com`

Table 8. Installation syntax for agent installation with agents in different network zones (continued)

Dynamic Agent workstation	Installation syntax

Dynamic Agent 4	`twsinst -new -uname <user_name>`
	`  -password user_password`
	`  -acceptlicense yes`
	`  -agent dynamic`
	`  -gateway local`
	`  -gwid GWADMIN`
	`  -jmport 31118`
	`  -tdwbport 31116`
	`  -tdwbhostname mainbroker.mycompany.com`

Dynamic Agent 5	`twsinst -new -uname <user_name>`
	`  -password user_password`
	`  -acceptlicense yes`
	`  -agent dynamic`
	`  -gateway remote`
	`  -tdwbport 31118`
	`  -tdwbhostname agent4.mycompany.com`

where,

Dynamic agent 3

-gateway local

Indicates that Dynamic Agent 3 can communicate with the dynamic workload broker directly, and a gateway is installed on Dynamic Agent 3 to route communications from dynamic agent workstations that cannot directly communicate with the dynamic workload broker.

-gwid GWADMIN

The gateway ID, GWADMIN, is the name that identifies the gateway on Dynamic Agent 3. Gateways with the same gateway_id can mutually take over in routing communications to the agents connected to them. Specify a different <*gateway_id*> if the gateways do not communicate with each other.

In addition, configure the two gateways in parallel to take over routing communications from the agents connected to them, should one of the gateways become unavailable. Edit the JobManagerGW.ini file on Dynamic agent 3 and set the JobManagerGWURIs property as follows:

```
JobManagerGWURIs = https://agent3.mycompany.com:31118/ita/JobManagerGW/
 JobManagerRESTWeb/JobScheduler/resource,https://agent4.mycompany.com:
 31118/ita/JobManagerGW/JobManagerRESTWeb/JobScheduler/resource
```

-tdwbport 31116

The port number of the dynamic workload broker.

-tdwbhostname mainbroker.mycompany.com

The fully qualified host name of the dynamic workload broker.

Dynamic agent 4

-gateway local

Indicates that Dynamic Agent 4 can communicate with the dynamic workload broker directly, and a gateway is installed on Dynamic Agent 4 to route communications from dynamic agent workstations (Dynamic agent 5) that cannot directly communicate with the dynamic workload broker.

-gwid GWADMIN

The gateway ID, GWADMIN, is the name that identifies the gateway site on Dynamic Agent 4. Gateways with the same <*gateway_id*> can mutually take over in routing communications to the agents connected to them. Specify a different <*gateway_id*> if the gateways do not communicate with each other.

In addition, you can configure the two gateways in parallel to take over routing communications from the agents connected to them, should one of the gateways become unavailable. Edit the JobManagerGW.ini file on Dynamic agent 4 and set the JobManagerGWURIs property as follows:

```
JobManagerGWURIs = https://agent3.mycompany.com:31118/ita/JobManagerGW/
 JobManagerRESTWeb/JobScheduler/resource,https://agent4.mycompany.com:
 31118/ita/JobManagerGW/JobManagerRESTWeb/JobScheduler/resource
```

-tdwbport 31116

The port number of the dynamic workload broker.

-tdwbhostname mainbroker.mycompany.com

The fully qualified host name of the dynamic workload broker.

Dynamic agent 5

-gateway remote

Indicates that Dynamic Agent 5 can connect to the internet through a gateway installed on a different agent, Dynamic Agent 4.

-tdwbport 31118

The port number of the dynamic agent workstation where the gateway resides. In this example, the port number of Dynamic Agent 4 is 31118.

-tdwbhostname agent4.mycompany.com

The fully qualified host name of the dynamic agent workstation where the gateway resides and to which the agent connects.

For information about configuring dynamic agents in this context see Configuring dynamic agent communications through a gateway *(on page)*.

Configuring a fault-tolerant agent

After installing a fault-tolerant agent, define the workstation in the database and link the workstation from the master. You can perform this task by using the Dynamic Workload Console or the command line interface. For information, see *User's Guide and Reference*. The following is an example of how to configure a fault-tolerant agent after installation using the command line interface:

1. Log in to the master domain manager as *TWS_user*.
2. Set the environment variables by running tws_env.sh.
3. Create the workstation definition in the IBM Workload Scheduler database. Open a command line window and enter the following commands:

   ```
   composer
   new
   ```

4. Type the workstation definition in the text editor. For example:

   ```
   CPUNAME F235007_00
      DESCRIPTION "fault-tolerant agent"
      OS UNIX
      NODE lab235007
      TCPADDR 31111
      DOMAIN MASTERDM
      FOR MAESTRO
        TYPE FTA
        AUTOLINK ON
        BEHINDFIREWALL OFF
        FULLSTATUS OFF
   END
   ```

   Run JnextPlan with the option **-for 0000** to add the agent workstation definition to the plan and to send the Symphony file to it. For more information about workstation definitions, see Workstation definition *(on page)*.

Note: Ensure that the global option carryforward is set to all, otherwise only incomplete job streams are carried forward.

5. If you set the autolink parameter to OFF, issue the link command from the master domain manager to link the agent and to download the Symphony file to it:

```
conman “link workstation“
```

6. Change the workstation limit to allow jobs to run on the workstation. For example, set the number of jobs to run concurrently on the workstation to 10:

```
conman "limit F235007_00;10"
```

Additionally, the following configuration procedures might be necessary:

- Customizing and configuring global, local, and user options. See the relevant sections in Customizing and configuring IBM Workload Scheduler *(on page)*
- Customizing and configuring user authentication to allow users authorization on actions and objects, and to configure LDAP. See Configuring authentication *(on page)*.
- Setting connection security to enable SSL for inter-component communications. See the relevant sections in Connection security overview *(on page)*.

Installing additional IBM Workload Scheduler components

This section describes how to install additional IBM Workload Scheduler components.

If you need to install more IBM® Workload Scheduler components, for example if you need to add an additional component to an existing installation, you can perform the steps described in the relevant topic:

- Installing an additional backup domain manager *(on page 99)*
- Installing dynamic domain components *(on page 101)*
- Installing agents on IBM i systems *(on page 110)*

Installing an additional backup domain manager

Considerations about installing an additional backup domain manager

You can perform a typical installation, as described in the following scenario, or you can customize the installation parameters, as described in FAQ - master domain manager and backup master domain manager customizations *(on page 74)*.

The backup domain manager shares the database with its master domain manager and requires a dedicated WebSphere Application Server Liberty Base, installed on the same workstation as the backup domain manager.

After installing a master domain manager, the administrator runs the **serverinst** command again to install a backup domain manager on a dedicated workstation. The backup domain manager is an agent that can assume the responsibilities of its master domain manager. The **serverinst** command connects to the database you specify, discovers that a master domain manager is already installed, and proceeds to install a backup domain manager.

You might want to install an additional backup domain manager for increased performance and reliability, for example you can move the event processor or the Dynamic Workload Console workload to the backup domain manager.

The IBM® Workload Scheduler administrator needs the following information, which is the same provided when installing the master domain manager, with the exception of the WebSphere Application Server Liberty Base installation directory, which is located on the workstation where you are installing the backup domain manager:

Table 9. Required information

Required information for performing the installation

Command parameter	Information type	Provided in...
Database information		
--rdbmstype	database type	Creating and populating the database *(on page 40)*
--dbhostname	database hostname	
--dbport	database port	
--dbname	database name	
--dbuser	database user name	
--dbpassword	database password	
IBM® Workload Scheduler information		
--wauser	IBM® Workload Scheduler administrative user name	Creating the IBM Workload Scheduler administrative user *(on page 69)*
--wapassword	IBM® Workload Scheduler administrative user password	
WebSphere Application Server Liberty Base information		
--wlpdir	WebSphere Application Server Liberty Base installation directory	Installing WebSphere Application Server Liberty Base *(on page 37)*

Before starting the backup domain manager installation, ensure the following steps have been completed:

1. Installing WebSphere Application Server Liberty Base *(on page 37)* on the workstation where you plan to install the backup domain manager.
2. Encrypting passwords (optional) *(on page 39)*.
3. Creating and populating the database *(on page 40)* for the master domain manager. The backup domain manager shares the database with the master domain manager.
4. Creating the IBM Workload Scheduler administrative user *(on page 69)*
5. Installing the master domain manager and backup master domain manager *(on page 70)*
6. On UNIX™ operating systems, ensure that **umask** is set to **022**. To verify that **umask** is set to the correct value, from a command prompt, run the **umask** command. If the value is different from **022**, modify it by running the following command:

   ```
   umask 022
   ```

You can run the **serverinst** command specifying a typical set of parameters. In this case, default values are used for all remaining parameters. For more information about all **serverinst** parameters and default values, see Server components installation - serverinst script *(on page 310)*.

A properties file named `serverinst.properties` is available if you do not want to type parameters in the command line. This is expecially useful if you need to specify many parameters or if you want to reuse the file for several installations. The file is located in *`image_location`*`/TWS/`*`interp_name`*.

You can specify values in the properties file, type them in the command line, or use both methods. If a parameter is specified both in the properties file and in the command line, the command line value takes precedence.

If you need to modify any of the default values, edit the `serverinst.properties` file, but do not modify the `serverinst.template` file located in the same path.

To install the backup domain manager, perform the following steps:

1. Log in to the workstation where you plan to install as root.
2. Browse to the folder where the `serverinst` command is located in *`image_location`*`/TWS/`*`interp_name`*.
3. . Start the installation specifying a minimum set of parameters. In this case, default values are used for all remaining parameters:

   On Windows operating systems

   ```
   cscript serverinst.vbs --acceptlicense yes --rdbmstype <db_type>
       --dbhostname <db_hostname> --dbport <db_port> --dbname <db_name>
       --dbuser <db_user> --dbpassword <db_password> --wauser <wa_user>
       --wapassword <wa_password> --wlpdir <Liberty_installation_dir>
   ```

   On UNIX operating systems

   ```
   ./serverinst.sh --acceptlicense yes --rdbmstype <db_type>
           --dbhostname <db_hostname> --dbport <db_port> --dbname <db_name>
           --dbuser <db_user> --dbpassword <db_password> --wauser <wa_user>
           --wapassword <wa_password> --wlpdir <Liberty_installation_dir>
   ```

4. To verify that the installation completed successfully, browse to the directory where you installed the backup domain manager and type the following commands:

   ```
   . ./tws_env.sh
   ```

   ```
   optman ls
   ```

   This command lists the IBM® Workload Scheduler configurations settings and confirms that IBM® Workload Scheduler installed correctly.

You have now successfully installed the backup domain manager.

If you want to customize more installation parameters, see FAQ - master domain manager and backup master domain manager customizations *(on page 74)*.

Installing dynamic domain components

Procedure to install the dynamic domain manager and backup dynamic domain manager

A dynamic domain manager is the management hub in a domain running both static and dynamic workload. All communications to and from the dynamic agents in the domain are routed through the dynamic domain manager.

The following topics describe the required steps:

1. Installing WebSphere Application Server Liberty Base *(on page 101)*
2. Encrypting passwords (optional) *(on page 102)*
3. Creating and populating the database for the dynamic domain manager *(on page 104)*
4. Creating the IBM Workload Scheduler administrative user *(on page 105)*
5. Installing the dynamic domain manager and backup dynamic domain manager *(on page 106)*

Installing WebSphere Application Server Liberty Base

WebSphere Application Server Liberty Base is required on all workstations where you plan to install the server components and the Dynamic Workload Console.

Ensure that your system meets the operating system and Java requirements. For more information, see WebSphere Application Server Liberty Base detailed system requirements.

You can quickly install WebSphere Application Server Liberty Base by extracting an archive file on all supported platforms.

Install WebSphere Application Server Liberty Base on all of the following workstations, which comprise a typical installation:

- master domain manager
- backup domain manager
- two Dynamic Workload Console installations on two separate workstations

If you plan to install a dynamic domain manager and its backup, these components require a separate WebSphere Application Server Liberty Base installation.

To extract the archive, you can use your own Java Ext or use the Java Ext provided with the IBM® Workload Scheduler image. The provided Java Ext is located in the following path in the image for your operating system: `<IMAGE_DIR>/TWS/<INTERP>/Tivoli_Eclipse_<INTERP>/TWS/JavaExt/`.

To install WebSphere Application Server Liberty Base, perform the following steps:

1. Find out which version of WebSphere Application Server Liberty Base is required, by running the Detailed Software Requirements report and browsing to the **Prerequisites** tab.
2. Download WebSphere Application Server Liberty Base from Recommended updates for WebSphere Application Server Liberty.
3. Install WebSphere Application Server Liberty Base by extracting the archive file to a directory of your choice.

   On Windows operating systems

   ```
   java -jar <liberty_download_dir>\wlp-base-all-<version>.jar
   --acceptLicense <install_dir>
   ```

   On UNIX operating systems

   ```
   ./java -jar <liberty_download_dir>/wlp-base-all-<version>.jar
   --acceptLicense <install_dir>
   ```

   where:

   <liberty_download_dir>

   The directory where you downloaded WebSphere Application Server Liberty Base.

   <version>

   The number of the version.

   <install_dir>

   The directory where you want to install WebSphere Application Server Liberty Base.

   Note: Note that the value of the *<install_dir>* parameter must match the value to be defined for the **wlpdir** parameter when installing the master domain manager and its backup, dynamic domain manager and its backup, and the Dynamic Workload Console.

4. Ensure the IBM® Workload Scheduler administrative user has the rights to run WebSphere Application Server Liberty Base and full access to the installation directory. If WebSphere Application Server Liberty Base is shared between the master domain manager and the Dynamic Workload Console, ensure also the Dynamic Workload Console user has the same rights.

You have now successfully installed WebSphere Application Server Liberty Base.

You can now proceed to Encrypting passwords (optional) *(on page 102)*.

Encrypting passwords (optional)

How to encrypt passwords required by the installation process

You can optionally encrypt the passwords that you will use while installing, upgrading, and managing IBM® Workload Scheduler. The secure command uses the AES method and prints the encrypted password to the screen or saves it to a file.

Note: It is important you understand the limits to the protection that this method provides. The custom passphrase you use to encrypt the passwords is stored in clear format in the `passphrase_variables.xml` file, stored in `configureDropin`. To fully understand the implications of this method, it is recommended you read the information provided by WebSphere Application Server Liberty Base at the link Liberty: The limits to protection through password encryption.

You can perform a typical procedure, which uses a custom passphrase, as described in the following scenario. For more information about all secure arguments and default values, see Optional password encryption - secure script *(on page 300)*.

Encrytping the password

1. Browse to the folder where the secure command is located:
   - Before the installation, the command is located in the product image directory, `<image_directory>/TWS/<op_sys>/Tivoli_LWA_<op_sys>/TWS/bin`
   - After the installation, the command is located in `TWA_home/TWS/bin`
2. Depending on your operating system, encrypt the password as follows:

   Windows operating systems

   ```
   secure -password password -passphrase passphrase
   ```

   UNIX operating systems

   ```
   ./secure -password password -passphrase passphrase
   ```

   z/OS operating systems

   ```
   ./secure -password password -passphrase passphrase
   ```

   where

   -password

   Specifies the password to be encrypted.

   -passphrase

   Optional. Specifies the custom passphrase that is used to generate the key with which the command encrypts the password. If you set this parameter, inform the user who installs IBM Workload Automation that they must define the **SECUREWRAP_PASSPHRASE** environment variable in the same shell from which they run the installation command, and set it to the same value as the **passphrase** argument. On Windows operating systems, the passphrase must be at least 8 characters long.
3. Provide both the encrypted password and custom passphrase to the user in charge of installing IBM Workload Automation. You can use encrypted passwords only in association with the specific passphrase used to encrypt them.

Installing with the encrypted password

The user in charge of installing IBM Workload Automation must set the **SECUREWRAP_PASSPHRASE** environment variable by performing the following steps:

1. Open a brand new shell session.
2. Ensure that no value is set for the **SECUREWRAP_PASSPHRASE** environment variable.
3. Define the **SECUREWRAP_PASSPHRASE** environment variable and set it to the passphrase defined by the user who ran the secure command, as follows:

   ```
   SECUREWRAP_PASSPHRASE=<passphrase>
   ```

You can use encrypted passwords only in association with the specific passphrase used to encrypt them.

4. In the same shell session, provide the encrypted passwords when running any command that uses a password. An encrypted password looks like the following example:

```
{aes}AFC3jj9cROYyqR+3CONBzVi8deLb2Bossb9GGroh8UmDPGikIkzXZzid3nzY0IhnSg=
```

You can now proceed to Creating and populating the database for the dynamic domain manager *(on page 104)*.

Creating and populating the database for the dynamic domain manager

Instructions for creating and populating the IBM® Workload Scheduler database for the dynamic domain manager

The procedure for creating the database for the dynamic domain manager is identical to that of the master domain manager, with the exception that an additional parameter, `component_type`, must be passed to the script.

For the complete procedure for creating and populating the database, see Creating and populating the database *(on page 40)*, then select the procedure related to the database you are using.

To create a DB2 database for the dynamic domain manager submit the following command:

On Windows operating systems

```
cscript configureDb.vbs --componenttype DDM --dbhostname db_hostname
    --dbport db_port --dbname db_name --dbuser db_user
    --dbadminuser db_administrator
    --dbadminuserpw db_administrator_password
```

On UNIX operating systems

```
./configureDb.sh --componenttype DDM --dbhostname db_hostname
     --dbport db_port --dbname db_name --dbuser db_user
     --dbadminuser db_administrator
     --dbadminuserpw db_administrator_password
```

To create an Oracle database for the dynamic domain manager submit the following command:

On Windows operating systems

```
cscript configureDb.vbs --componenttype DDM --rdbmstype ORACLE
    --dbname service_name --dbuser db_user --dbpassword db_password
    --dbhostname db_hostname --dbadminuser db_administrator
    --dbadminuserpw db_administrator_password
```

On UNIX operating systems

```
./configureDb.sh --componenttype DDM --rdbmstype ORACLE
    --dbname service_name --dbuser db_user --dbpassword db_password
    --dbhostname db_hostname --dbadminuser db_administrator
    --dbadminuserpw db_administrator_password
```

To create an MSSQL database for the dynamic domain manager submit the following command:

On Windows operating systems

```
cscript configureDb.vbs --componenttype DDM --rdbmstype MSSQL
    --dbname db_name --dbhostname db_hostname
    --dbadminuser db_administrator
    --dbadminuserpw db_administrator_password
```

On UNIX operating systems

```
./configureDb.sh --componenttype DDM --rdbmstype MSSQL
    --dbname db_name --dbhostname db_hostname
```

```
    --dbadminuser db_administrator
    --dbadminuserpw db_administrator_password
```

where:

--componenttype

The IBM® Workload Scheduler for which the database is installed. When installing a dynamic domain manager, specify **DDM**.

--dbhostname *db_hostname*

The host name or IP address of database server.

--dbport *db_port*

The port of the database server.

--dbname *db_name*

The name of the IBM® Workload Scheduler database. Note that this name must match the name specified in the `serverinst` command. For more information about the `serverinst` command, see Server components installation - serverinst script *(on page 310)*. When creating the database on Oracle, this parameter indicates the service name.

--dbuser *db_user*

The user that has been granted access to the IBM® Workload Scheduler tables on the database server.

--dbpassword *db_password*

(Oracle DB only) The password for the user that has been granted access to the IBM® Workload Scheduler tables on the database server. Special characters are not supported.

--dbadminuser*db_admin_user*

The database administrator user that creates the IBM® Workload Scheduler schema objects on the database server.

--dbadminuserpw *db_admin_password*

The password of the DB administrator user that creates the IBM® Workload Scheduler schema objects on the database server. Special characters are not supported.

The same criteria apply when creating the database for all supported databases. For more information about creating the database for each supported vendor, see:

- Creating and populating the database for DB2 for the master domain manager *(on page 41)*
- Creating the database for Oracle and Amazon RDS for Oracle for the master domain manager *(on page 47)*
- Creating the database for MSSQL for the master domain manager *(on page 50)*

You have now successfully created and populated the IBM® Workload Scheduler database.

You can now proceed to Creating the IBM Workload Scheduler administrative user *(on page 105)*.

Creating the IBM® Workload Scheduler administrative user

Instructions to create the IBM® Workload Scheduler administrative user

IBM® Workload Scheduler administrative user

The IBM® Workload Scheduler administrator creates the administrative user (**wauser**). The administrative user is the user for which the product will be installed in the subsequent steps. This implies that this user has full access to all scheduling objects.

The user name can contain alphanumeric, dash (-), and underscore (_) characters; it cannot contain national characters. The first character of the user name must be a letter.

The following considerations apply:

On Windows operating systems:

- If this user account does not already exist, it is automatically created at installation time.
- If installing on a Windows™ server in a domain, do not define a domain and local ID with the same user name.
- If you specify a domain user, define the name as *domain_name\user_name*.
- If you specify a local user, define the name as *system_name\user_name*. Type and confirm the password.

On UNIX and Linux operating systems:

This user account must be created manually before running the installation and must be enabled to login to the machine where the master domain manager is going to be installed. Create a user with a home directory and group. Use the appropriate UNIX and Linux operating system commands to create the user.

Important: Group names that contain a "/" (forward slash) character can cause permissions to not be set correctly. When IBM® Workload Scheduler retrieves credentials from WebSphere Application Server Liberty, it parses the returned list of groups names assuming they are saved in the format `<realm_name>/<group_name>`. If the group name, the realm name, or both contain a "/" character, the parsing fails.

You can also install IBM® Workload Scheduler using a user different from the root user. This installation method is known as **no-root installation** and applies to all IBM® Workload Scheduler components. Note that if you choose this installation method, only the user who performs the installation can use IBM® Workload Scheduler. For this reason, the typical installation scenario described in this section uses the root user.

For more information, see IBM Workload Scheduler user management *(on page 34)*.

Results

You have now successfully created the IBM® Workload Scheduler administrative user.

What to do next

You can now proceed to Installing the dynamic domain manager and backup dynamic domain manager *(on page 106)*.

Installing the dynamic domain manager and backup dynamic domain manager

Considerations about installing the dynamic domain manager and backup dynamic domain manager

A dynamic domain manager is the management hub in a domain running both static and dynamic workload. All communications to and from the dynamic agents in the domain are routed through the dynamic domain manager.

You can perform a typical installation, as described in the following scenario, or you can customize the installation parameters, as described in FAQ - dynamic domain manager customizations *(on page 108)*. For example, you can install the dynamic domain manager and backup dynamic domain manager using custom certificates.

The dynamic domain manager and backup dynamic domain manager require a dedicated database and a dedicated WebSphere Application Server Liberty Base.

The procedure to install the dynamic domain manager and backup dynamic domain manager is exactly the same, with the difference that it is performed on two different workstations and that each installation points to its local WebSphere Application

Server Liberty Base installation. IBM® Workload Scheduler determines whether or not a dynamic domain manager is already present in the environment and proceeds to install a dynamic domain manager or backup dynamic domain manager accordingly.

The IBM® Workload Scheduler administrator installs the dynamic domain manager and backup dynamic domain manager. He needs the following information:

Table 10. Required information

Required information for performing the installation

Command parameter	Information type	Provided in...
Database information		
--rdbmstype	database type	Creating and populating the database for the dynamic domain manager *(on page 104)*
--dbhostname	database hostname	
--dbport	database port	
--dbname	database name	
--dbuser	database user name	
--dbpassword	database password	
IBM® Workload Scheduler information		
--wauser	IBM® Workload Scheduler administrative user name	Creating the IBM Workload Scheduler administrative user *(on page 105)* and Installing the master domain manager and backup master domain manager *(on page 70)*
--wapassword	IBM® Workload Scheduler administrative user password	
--master	The master domain manager name	
--mdmbrokerhostname	The fully qualified host name or IP address of the master domain manager contacted by the dynamic domain manager.	
--mdmhttpsport	The port of the master domain manager host used by the broker to contact master domain manager.	
WebSphere Application Server Liberty Base information		
--wlpdir	WebSphere Application Server Liberty Base installation directory	Installing WebSphere Application Server Liberty Base *(on page 101)*

Before starting the installation, ensure the following steps have been completed:

1. Installing WebSphere Application Server Liberty Base *(on page 101)* on the workstation where you plan to install the dynamic domain manager and on the workstation where you plan to install the backup dynamic domain manager.
2. Encrypting passwords (optional) *(on page 102)*.
3. Creating and populating the database for the dynamic domain manager *(on page 104)*
4. Creating the IBM Workload Scheduler administrative user *(on page 105)*
5. On UNIX™ operating systems, ensure that **umask** is set to **022**. To verify that **umask** is set to the correct value, from a command prompt, run the **umask** command. If the value is different from **022**, modify it by running the following command:

   ```
   umask 022
   ```

You can run the **serverinst** command specifying a typical set of parameters. In this case, default values are used for all remaining parameters. For more information about all **serverinst** parameters and default values, see Server components installation - serverinst script *(on page 310)*.

A properties file named `serverinst.properties` is available if you do not want to type parameters in the command line. This is expecially useful if you need to specify many parameters or if you want to reuse the file for several installations. The file is located in *`image_location`*`/TWS/`*`interp_name`*.

You can specify values in the properties file, type them in the command line, or use both methods. If a parameter is specified both in the properties file and in the command line, the command line value takes precedence.

If you need to modify any of the default values, edit the `serverinst.properties` file, but do not modify the `serverinst.template` file located in the same path.

To install the dynamic domain manager, perform the following steps:

1. Log in to the workstation where you plan to install.
2. Browse to the folder where the `serverinst` command is located:

   On Windows operating systems

   `image_location``\TWS\`*`interp_name`*

   On UNIX operating systems

   `image_location``/TWS/`*`interp_name`*
3. Start the installation specifying a typical set of parameters. In this case, default values are used for all remaining parameters:

   On Windows operating systems

   ```
   cscript serverinst.vbs --acceptlicense yes --rdbmstype db_type
           --dbhostname db_hostname --dbport db_port --dbname db_name
           --dbuser db_user --dbpassword db_password --wauser wa_user
           --wapassword wa_password --componenttype DDM --domain domain_name
           --master mdm_name --mdmbrokerhostname mdm_broker_host_name
           --mdmhttpsport mdm_https_host_name --wlpdir Liberty_installation_dir\wlp
   ```

   On UNIX operating systems

   ```
   ./serverinst.sh --acceptlicense yes --rdbmstype db_type
           --dbhostname db_hostname --dbport db_port --dbname db_name
           --dbuser db_user --dbpassword db_password --wauser wa_user
           --wapassword wa_password --componenttype DDM --domain domain_name
           --master mdm_name --mdmbrokerhostname mdm_broker_host_name
           --mdmhttpsport mdm_https_host_name --wlpdir Liberty_installation_dir/wlp
   ```

Repeat the same procedure on the workstation where you plan to install the backup dynamic domain manager

Result

You have now successfully installed the dynamic domain manager and backup dynamic domain manager.

For more information about all **serverinst** parameters and default values, see Server components installation - serverinst script *(on page 310)*.

FAQ - dynamic domain manager customizations

A list of questions and answers related to the customization of the dynamic domain manager installation

When installing the dynamic domain manager, you can perform a typical installation, as described in Installing the dynamic domain manager and backup dynamic domain manager *(on page 106)* or you can customize a number of parameters, as described in the following scenario:

- How do I install the dynamic domain manager using custom certificates? *(on page 109)*

How do I install the dynamic domain manager using custom certificates?

Installing the dynamic domain manager and its backup using custom certificates

You can install the dynamic domain manager and its backup using default certificates, as described in Installing the dynamic domain manager and backup dynamic domain manager *(on page 106)*, or you can optionally use custom certificates.

To install dynamic domain manager and backup dynamic domain manager using custom certificates, perform the following steps:

1. Generate the custom certificates required for installing the dynamic domain manager and backup dynamic domain manager, as follows:

   ```
   openssl genrsa -des3 -out tls.key 2048
   ```

   The following files are created:
   - `ca.crt`
   - `tls.key`
   - `tls.crt`
2. Copy the files to a path of your choice on the workstation where you plan to install the dynamic domain manager or backup dynamic domain manager. When performing the installation, you provide this path using the sslkeysfolder parameter.
3. Copy the `tls.crt` file from the master domain manager to the workstation where you plan to install the dynamic domain manager or backup dynamic domain manager. Specify a different path from the path of the above certificates to avoid overwriting the existing `tls.crt`.
4. Rename the `tls.crt` file from the master domain manager to `jwt.crt`.
5. Copy the `jwt.crt` file to the same path as the certificates generated in step 1.
   You now have on the workstation where you plan to install the dynamic domain manager or backup dynamic domain manager four certificate files:
   a. `ca.crt`
   b. `tls.key`
   c. `tls.crt`
   d. `jwt.crt`
6. Browse to the folder where the `serverinst` command is located:

   On Windows operating systems

   image_location\TWS*interp_name*

   On UNIX operating systems

   image_location/TWS/*interp_name*
7. Start the installation specifying the path to the dynamic domain manager certificates using the sslkeysfolder parameter:

   On Windows operating systems

   ```
   cscript serverinst.vbs --acceptlicense yes --rdbmstype db_type
           --dbhostname db_hostname --dbport db_port --dbname db_name
           --dbuser db_user --dbpassword db_password --wauser wa_user
           --wapassword wa_password --componenttype DDM --domain domain_name
           --master mdm_name --mdmbrokerhostname mdm_broker_host_name
           --mdmhttpsport mdm_https_host_name --wlpdir Liberty_installation_dir\wlp
           --sslkeysfolder path_to_certificates --sslpassword certificate_password
   ```

   On UNIX operating systems

   ```
   ./serverinst.sh --acceptlicense yes --rdbmstype db_type
           --dbhostname db_hostname --dbport db_port --dbname db_name
           --dbuser db_user --dbpassword db_password --wauser wa_user
           --wapassword wa_password --componenttype DDM --domain domain_name
           --master mdm_name --mdmbrokerhostname mdm_broker_host_name
           --mdmhttpsport mdm_https_host_name --wlpdir Liberty_installation_dir/wlp
           --sslkeysfolder path_to_certificates --sslpassword certificate_password
   ```
8. Repeat the same procedure for the backup dynamic domain manager.

You have now successfully installed the dynamic domain manager and backup dynamic domain manager

Installing agents on IBM i systems

Learn how to install agents on IBM i systems.

To install dynamic agents and z-centric agents on IBM i systems, use the `twsinst` installation script.

You can either use the default **QSECOFR** user or create a new user with **ALLOBJ authority**. If you plan to use a user different from **QSECOFR**, create the user before the installation and assign it the ALLOBJ authority.

For dynamic agents, you can also use a user **different from QSECOFR** with no specific authority. Note that only the user who performs the installation can use the agent.

Verify that the user profile used as **TWSUser** is not a member of a group profile. Set the group profile associated with the **TWSUser** to **NONE*. If the **TWSUser** is a member of a group, the installation might fail.

To install the agents, perform the following steps:

1. Sign on as the user of your choice, either **QSECOFR** or an **existing user with ALLOBJ authority**. If you use a user different from **QSECOFR**, specify the **allObjAuth** parameter to indicate that the specified user has the ALLOBJ authority. Ensure the user is existing and has ALLOBJ authority because the product does not verify that the correct authority is assigned. For more information about the **allObjAuth** parameter, see Agent installation parameters on IBM i systems *(on page 112)*. Only for dynamic agents, you can also use a user different from **QSECOFR** with no specific authority.
2. Create an IBM i user profile for which the IBM Workload Scheduler agent is installed.

   Note: The user profile is not the same as for the user that is performing the installation, unless you use a user different from **QSECOFR** with no specific authority (dynamic agents only). Instead the user profile is for the user that you specify in the **-uname** *username* parameter when running the twsinst script. For descriptions of the syntax parameters, see Agent installation parameters on IBM i systems *(on page 112)*. You cannot use an existing IBM i system user profile, an application supplied user profile, or any of the following reserved IBM i user profiles:
   - QDBSHR
   - QDFTOWN
   - QDOC
   - QLPAUTO
   - QLPINSTALL
   - QRJE
   - QSECOFR
   - QSPL
   - QSYS
   - QTSTRQS

   Attention: Consider that:
   - If the user profile is a member of a group, the installation fails. Set the group profile that is associated with the user profile to **NONE*.
   - If the *username* is longer than 8 characters, after the installation the agent (and the JobManager component) runs under the **QSECOFR** user instead of under the authority of the installation user. To prevent this problem, set the `PASE_USRGRP_LIMITED` environment variable to N.

3. On the IBM i system, verify that no library exists with the same name as the user profile supplied for the agent user.
4. Download the installation images from IBM Fix Central.
5. To untar or unzip the agent eImage, you can use the *PASE* shell or the *AIXterm* command.

   Using *PASE* shell:

   a. Open the *PASE* shell.
   b. Run the command "CALL QP2TERM".
   c. Locate the folder where you downloaded the agent eImage and run the command:

IBM Z Workload Scheduler Agent

```
"tar xvf TWSversion_number>_IBM_I.tar"
```

Dynamic Agent

```
"unzip TWSversion_number>_IBM_I.zip"
```

d. Exit from the *PASE* shell.

Using *AIXterm* command:

a. Start the *Xserver* on your desktop.
b. On the iSeries machine, open a *QSH shell* and export the display.
c. In *QSH shell*, go to the directory */QopenSys* and run the command "aixterm -sb".
d. A pop-up window is displayed on your desktop. By Using this pop-up window, unzip the *TWSversion_number>_IBM_I.zip* file, or untar the *TWSversion_number>_IBM_I.tar* file.

6. If your machine's primary language is other than English, carry out these steps:
   a. Add English as secondary language.
   b. Ensure that when connecting to the environment the Host Code-Page is set to 037
   c. Before starting the installation, verify that the *Qshell session* is configured correctly and type the following command in the <yourfilename> :

   ```
   echo " key key2 " | sed 's/ *$//g' | sed 's/^ *//g'
   ```

   d. Run the <yourfilename>
   e. The environment is configured in the correct way if the output is: "key key2".

7. Open a *QSH shell* and run the twsinst script. During the installation process, the product creates an IBM i library and a job description with the same name as the user profile created in Step 2 *(on page 110)*.

   The installation procedure adds this library to the user profile library list of the dynamic agent user profile and sets this job description as the job description of the dynamic agent user profile. By default, the software is installed in the user's home directory.

If the installation fails to understand the cause of the error, see Analyzing return codes for agent installation, upgrade, restore, and uninstallation *(on page 280)*.

After a successful installation, perform the following configuration task:

- Configuring a dynamic agent *(on page 145)*.

Command usage and version

Show command usage and version

```
twsinst -u | -v
```

Install a new instance

```
twsinst -new -uname username
   -acceptlicense yes|no
   [-addjruntime true|false]
   [-agent dynamic]
   [-allObjAuth]
   [-company company_name]
   [-displayname agentname]
   [-gateway local|remote|none]
   [-gweifport gateway_eif_port]
   [-gwid gateway_id]
   [-hostname hostname]
   [-inst_dir install_dir]
   [-jmport port_number]
   [-jmportssl true|false]
   [-lang lang_id]
   -tdwbport tdwbport_number
```

```
-tdwbhostname host_name
[-work_dir working_dir]
```

For a description of the installation parameters and options that are related to agent on this operating system, see Agent installation parameters on IBM i systems *(on page 112)*.

Prerequisites

To install and use the IBM i agent you must have a supported version of the IBM i operating system. For a detailed list of supported operating systems, see the Detailed System Requirements document at IBM Workload Scheduler Detailed System Requirements.

Scanning system prerequisites on IBM i systems

Scanning system prerequisites on IBM i systems

Before you install or upgrade the agent, IBM Workload Scheduler automatically runs a scan on your system. Having an environment that meets the product system requirements ensures that the installation or upgrade succeeds without any delays or complications.

The scan verifies that:

- The operating system is supported for the product.
- There is enough permanent and temporary disk space to install both the product and its prerequisites.
- There is enough memory and virtual memory swap space.

Note: The scan verifies only that the environment meets the requirements of IBM Workload Scheduler.

If any of these checks fails, IBM Workload Scheduler performs the following action:

- An error message is returned. Analyze the log file, solve the error, and rerun the installation or upgrade. The log file is in *%TEMP%*`\TWA\tws1023\result.txt`
- You can decide to rerun the installation or upgrade without executing the prerequisite scan. If you specify the **-skipcheckprereq** parameter, the twsinst installation script does not execute the prerequisite scan. For more information about the `-skipcheckprereq` option, see Agent installation parameters - twsinst script *(on page 84)*.

For a detailed list of supported operating systems and product prerequisites, see IBM Workload Scheduler Detailed System Requirements.

Agent installation parameters on IBM i systems

The parameters set when using the **twsinst** script to install dynamic and z-centric agents on IBM i systems.

-acceptlicense *yes*|*no*

Specify whether to accept the License Agreement.

-addjruntime *true*|*false*

Adds the Java™ run time to run job types with advanced options, both those types that are supplied with the product and the additional types that are implemented through the custom plug-ins. Valid values are **true** and **false**. The default for a fresh installation is **true**. Set this parameter to `true` if you use the **sslkeysfolder** and **sslpassword** parameters to define custom certificates in PEM format.

If you decided not to install Java™ run time at installation time, you can still add this feature later as it is described in Adding a feature *(on page 152)*.

-allObjAuth

If you are installing, upgrading, or uninstalling with a user different from the default **QSECOFR** user, this parameter specifies that the user has the required ALLOBJ authority. Ensure the user is existing and has ALLOBJ authority because the product does not verify that the correct authority is assigned. The same user must be specified when installing, upgrading or uninstalling the agent. If you are using the **QSECOFR** user, this parameter does not apply.

-company ***company_name***

The name of the company. The company name cannot contain blank characters. The name is shown in program headers and reports. If not specified, the default name is COMPANY.

-displayname ***display_name***

The name to assign to the agent. The name cannot start with a number. The default is based on the host name of this computer.

If the host name starts with a number, the **-displayname** parameter must be specified.

-gateway ***local*|*remote*|*none***

Specifies whether to configure a gateway to communicate with the dynamic workload broker or not, and how it is configured. Specify `local` if the gateway is local to the dynamic agent workstation. Specify `remote` if the dynamic agent communicates through a gateway that is installed on a different dynamic agent workstation from the dynamic agent being installed. The default value is `none`, which means no gateway is configured. For information about installing with a local and remote gateway, see Example installation commands *(on page 92)*.

-gweifport ***gateway_eif_port***

Specifies the Job Manager Event Integration Facility (EIF) port number. The default value is **31132**. The valid range is 1 to 65535.

-gwid ***gateway_id***

The unique identifier for the gateway. This parameter is required when you specify **-gateway** `local`. The default gateway identifier that is assigned is **GW1**. The gateway identifier must start with either an alphabetic character or an underscore character (_), and it can contain only the following types of characters: alphabetic, numeric, underscores (_), hyphens (-), and periods (.).

Gateways can also work in parallel to mutually take over in routing communications to the agents connected to them. To enable gateways to work in parallel, all gateways must have the same *gateway_id* assigned. This information is stored in the `JobManagerGW.ini` file, by setting the JobManagerGWURIs property.

-hostname ***host_name***

The fully qualified hostname or IP address on which the agent is contacted by the dynamic workload broker. The default is the hostname of this computer. If the hostname is a localhost, the hostname parameter must be specified.

-inst_dir ***installation_dir***

The directory of the IBM Workload Scheduler installation. Specify an absolute path. The path cannot contain blanks. If you do not manually specify a path, the path is set to the default home directory, that is, the *home/username* directory, where *username* is the value specified in the -uname option.

-jmport ***port_number***

The JobManager port number used by the dynamic workload broker to connect to the dynamic agent. The default value is **31114**. The valid range is from 1 to 65535.

-jmportssl ***true*|*false***

The JobManager port used by the dynamic workload broker to connect to the IBM Workload Scheduler dynamic agent. The port value is the value of the ssl_port parameter in the `ita.ini` file if **-jmportssl** is set to `true`. If set to `false`, it corresponds to the value of the **tcp_port** parameter in the `ita.ini` file. The `ita.ini` file is located in `ITA\cpa\ita` on Windows™ systems and `ITA/cpa/ita` on UNIX™, Linux™, and IBM i systems.

Set the value to "true" if **- gateway** is set to `local`.

For communication using SSL or HTTPS

Set **jmportssl = *true***. To communicate with the dynamic workload broker, it is recommended that you set the value to *true*. In this case, the port specified in **jmport** communicates in HTTPS.

For communication without using SSL or through HTTP

Set **jmportssl = *false***. In this case the port specified in **jmport** communicates in HTTP.

-lang *lang_id*

The language in which the twsinst messages are displayed. If not specified, the system LANG is used. If the related catalog is missing, the default C language catalog is used. If neither **-lang** nor LANG are used, the default codepage is set to SBCS. For a list of valid values for these variables, see the following table:

Table 11. Valid values for -lang and LANG parameter

Language	Value
Brazilian portuguese	pt_BR
Chinese (traditional and simplified)	zh_CN, zh_TW
English	en
French	fr
German	de
Italian	it
Japanese	ja
Korean	ko
Russian	ru
Spanish	es

Note: This is the language in which the installation log is recorded and not the language of the installed engine instance. twsinst installs all languages as default.

-new

A fresh installation of the agent. Installs an agent and all supported language packs.

-skip_usercheck

Enable this option if the authentication process within your organization is not standard, thereby disabling the default authentication option. If you specify this parameter, you must create the user manually before running the script.

-skipcheckprereq

If you specify this parameter, IBM Workload Scheduler does not scan system prerequisites before installing the agent.

For a detailed list of supported operating systems and product prerequisites, see IBM Workload Scheduler Detailed System Requirements.

-sslkeysfolder

The name and path of the folder containing the certificates in PEM format. The installation program generates the keystore and truststore files using the password you specify with the **--sslpassword** parameter. If you use this parameter, ensure that the **addjruntime** parameter is set to true, because Java™ run time is required for defining custom certificates.

-sslpassword

Specify the password for the certificates automatically generated by the installation program. If you use this parameter, ensure that the **addjruntime** parameter is set to true, because Java™ run time is required for defining custom certificates.

-tdwbhostname *host_name*

The fully qualified host name of the dynamic workload broker. It is used together with the **-agent** *dynamic* and the **-tdwbport** *tdwbport_number* parameters. This value is registered in the **ResourceAdvisorUrl** property in the `JobManager.ini` file. This parameter is required.

If you set the **-gateway** parameter to `remote`, this is the host name of the dynamic agent hosting the gateway and to which the agent you are installing will connect. This information is stored in the `JobManager.ini` file. For information about installing with a local and remote gateway, see Example installation commands *(on page 92)*.

-tdwbport ***tdwbport_number***

The dynamic workload broker HTTP or HTTPS transport port number. It is used together with the **-agent** dynamic and the **-tdwbhostname** *host_name* parameters. The valid range is from 0 to 65535. If you specify **0**, you cannot run workload dynamically. Do not specify **0** if the *-agent* value is **dynamic**. This number is registered in the **ResourceAdvisorUrl** property in the `JobManager.ini` file. This parameter is required. .

If you set the **-gateway** parameter to `remote`, this is the HTTP or HTTPS port number of the dynamic agent hosting the gateway and to which the agent you are installing will connect. You have specified this port with the **jmport** parameter when installing the agent hosting the gateway. For information about installing with a local and remote gateway, see Example installation commands *(on page 92)*.

-thiscpu ***workstation***

The name of the IBM Workload Scheduler workstation of this installation. The name cannot exceed 16 characters, cannot start with a number, cannot contain spaces, and cannot be the same as the workstation name of the master domain manager. This name is registered in the `localopts` file. If not specified, the default value is the host name of the workstation.

If the host name starts with a number, **-thiscpu** parameter must be specified.

-u

Displays command usage information and exits.

-uname ***username***

The name of the user for which IBM Workload Scheduler is installed.

If you are using the **QSECOFR** user or a user with **ALLOBJ authority**, this user name is not the same as the user performing the installation. If you are using a user **different from QSECOFR**, the user performing the installation and the user for which the agent is installed are the same.

If *username* is longer than 8 characters, after installation the agent (and the JobManager component) erroneously run under the **QSECOFR** user, instead of under the authority of the installation user. To prevent this, set the `PASE_USRGRP_LIMITED` environment variable to N.

-work_dir ***working_dir***

The temporary directory used for the IBM Workload Scheduler installation process files deployment. The path cannot contain blanks. If you do not manually specify a path, the path is set to `/tmp/TWA/tws`*`version_number`*`>`.

-v

Displays the command version and exits.

Example installation of an agent on IBM i systems

The following example shows the syntax used when using the **twsinst** script to install a new instance of the agent on an IBM i system.

```
./twsinst -new
-uname TWS_user
-acceptlicense yes
-hostname thishostname.mycompany.com
-jmport 31114
-tdwbport 41114
```

```
   -tdwbhostname mainbroker.mycompany.com
   -work_dir "/tmp/TWA/tws93"
```

The twsinst script log files on IBM i systems

The twsinst log file name is:

Where: <*TWS_INST_DIR*>/twsinst_IBM_i_*TWS_user*^*product_version*.log

TWS_INST_DIR

The IBM Workload Scheduler installation directory. The default installation directory is /home/TWS_user.

TWS_user

The name of the user for which IBM Workload Scheduler was installed, that you supplied during the installation process.

product_version

Represents the product version. For example, for version 10.2.3 of the product, the value is 10.2.3.00

Chapter 2. Deploying with containers

Deploy IBM Workload Automation quickly and easily with containers.

Following you can find more details about the IBM Workload Automation deployment with containers based on your environment.

Docker containers

An easy and fast deployment method of IBM Workload Automation. Docker compose is a method to instantly download the product image, create a container, and start up the product.

Docker is a state-of-the-art technology which creates, deploys, and runs applications by using containers. Packages are provided containing an application with all of the components it requires, such as libraries, specific configurations, and other dependencies, and deploy it in no time on any other Linux or Windows workstation, regardless of any different settings between the source and the target workstation.

Docker adoption ensures standardization of your workload scheduling environment and provides an easy method to replicate environments quickly in development, build, test, and production environments, speeding up the time it takes to get from build to production significantly. Install your environment using Docker to improve scalability, portability, and efficiency.

Docker containers are available for UNIX, Windows and Linux on Z operating systems.

For more information, see the introductory readme file for all components available at IBM Workload Automation. You can also find detailed information for each component in the related readme file, as follows:

- IBM Workload Automation Server
- IBM Workload Automation Console
- IBM Workload Automation dynamic agent
- IBM Workload Automation z-centric agent

You can also use docker containers to store all the latest integrations available on Automation Hub. For further information see: Container plug-in.

Amazon Web Services (AWS) Elastic Kubernetes Service (EKS) (Amazon EKS)

You can use Amazon EKS to run IBM® Workload Scheduler containerized product components on the Amazon Web Services secure cloud platform.

For more information, see Deploying on Amazon EKS *(on page 124)*

Azure Kubernetes Service (AKS)

Deploy and manage IBM® Workload Scheduler containerized product components on the Azure AKS, a container orchestration service available on the Microsoft Azure public cloud. You can use Azure AKS to deploy, scale up, scale down and manage containers in the cluster environment. You can also deploy and run an Azure SQL database.

For more information, see Deploying on Azure AKS *(on page 124)*.

Google GKE

Google Kubernetes Engine (GKE) provides a managed environment for deploying, managing, and scaling your containerized applications using Google infrastructure. The Google GKE environment consists of multiple machines grouped together to form a cluster. You can also deploy and run Google Cloud SQL for SQL server.

Google GKE supports session affinity in a load balancing cluster, a feature which maintains each user session always active on the same pod. This ensures that the Dynamic Workload Console always connects to the same server during a session and that the user can perform any number of operations smoothly and seamlessly.

For more information, see Deploying on Google GKE *(on page 125)*.

Red Hat OpenShift

You can deploy the IBM Workload Automation components following one of the below procedures:

Deploying IBM Workload Automation components using helm charts

You can deploy the IBM Workload Automation components using IBM® certified containers. For further information, see Deploying IBM Workload Automation components using helm charts *(on page 123)*.

Deploying IBM Workload Automation components using IBM® Workload Automation Operator

You can deploy the IBM® Workload Automation Operator on your Red Hat OpenShift cluster first, and then use the operator to install the IBM Workload Automation components: the IBM Workload Automation server (master domain manager), Dynamic Workload Console, and the dynamic agent. IBM® certified containers are provided for the operator and for each of the product components. For more information, see Deploying IBM Workload Automation components using IBM Workload Automation Operator *(on page 124)*.

Considerations about deploying with containers

Some considerations about your IBM Workload Automation environments when the product components are deployed using containers.

An environment deployed with containers has some characteristics that differ from an environment installed using the classic installation method. Following a list of its characteristics:

- Container deployment is supported only for dynamic agents and not for fault-tolerant agents, and external fault-tolerant agents are not supported on Kubernetes.
- All dynamic agents must obligatorily be configured to use a dynamic agent gateway.
- Each time a switcheventprocessor command is issued, a switchmgr command must also be issued on the same node.
- An on-premises fault-tolerant agent cannot connect to a master domain manager for on-cloud solutions supported by IBM Workload Scheduler (only dynamic agents are supported).
- The IBM Workload Scheduler event processor service must run on the same machine where the current master is running because the host and port are re-mapped by dynamic agents to use the master server host and port. Thus, performing a switch from master to backup master, you must also switch the event processor on the new master.
- The console, server and agent components are installed with non-root user (`wauser`) that does not include sudoers privileges. This implies that jobs that run on agents on containers can run only with `wauser` user and cannot impersonate other users.
- An extended agent component, (*`RELEASE_NAME`*`-waserver-0_XA`), is automatically created on the server. It starts the scheduling process by running the FINAL job stream that generates the daily production plan.
- An FTA container is not provided (only dynamic agents are supported in containers).
- By default, the FINAL job stream has a start time of 07:00 and invokes MAKEPLAN at 07:00. The Start of Day is 00:00. MAKEPLAN extends the plan until 09:00 the following day. If you modify the scheduling time of the FINAL job stream to a time different from the default, then evaluate whether you should also manually modify the plan extension time defined in the MAKEPLAN job accordingly.

  The following is an example of the default output when you run `planman showinfo`:

```
Locale LANG set to the following: "en"
Plan creation start time: 06/23/2020 00:00 TZ UTC
Production plan start time of last extension: 06/24/2020 09:00 TZ UTC
Production plan end time: 06/25/2020 08:59 TZ UTC
Production plan time extention: 024:00
Plan last update: 06/24/2020 07:00 TZ UTC
Preproduction plan end time: 07/08/2020 00:00 TZ UTC
Start time of first not complete preproduction plan job stream instance: 06/23/2020 00:00 TZ UTC
```

Customizing container parameters

The document describes how to avoid the overwriting of the customized parameters added in the container configuration files, such as the `datasource.xml` file.

It is possible to customize the container configuration by adding parameters, for example, in the `datasource.xml` file that is located in the following path:

```
/opt/wautils/dropins
```

Restarting a container, the `datasource.xml` file is overwritten and the customized parameters inside it are lost; to avoid that, proceed as follows:

- Create another `.xml` file with a name that is listed in a higher alphabetical order than *datasource.xml*.
- In the new `.xml` file, add the parameters to be customized together with the corresponding section.

In this way - at the restart of the container - the customized parameters are not overwritten.

Deploying with Docker compose

Getting started with Docker compose

This topic gives you an overview of the high-level procedure to deploy IBM Workload Automation components using Docker.

To deploy IBM Workload Automation using a Docker container, proceed as follows:

1. Ensure that all of the prerequisites are met as documented in Prerequisites *(on page 119)*. If you are deploying on Linux on Z, ensure you perform the preparatory steps documented in Deploying Docker compose on Linux on Z *(on page 120)*.
2. Access and then download the Docker image from the entitled registry. For further information, see the complete procedure in Deploying containers with Docker *(on page 121)*.
3. You can choose to deploy all product containers with a single command, or you can deploy each product component container individually.

   For more information, see the introductory readme file for all components available at IBM Workload Automation. You can also find detailed information for each component in the related readme file, as follows:
   - IBM Workload Automation Server
   - IBM Workload Automation Console
   - IBM Workload Automation dynamic agent
   - IBM Workload Automation z-centric agent
4. Access the container to verify the status and run IBM Workload Automation commands. For further details see Accessing the Docker containers *(on page 122)*.

Prerequisites

Prerequisite information when deploying with containers.

When deploying the product using containers, ensure you have fulfilled the following prerequisites:

Check the prerequisites of the command-line installation method in Prerequisites *(on page 31)*.

If you want to calculate the necessary resources that the agent container needs to run, use the following formula:

Evaluate the volume_size variable:

```
Volume size(MB)=
        120 + [ 30 x jobs_per_day x (average_joblog_size_MB / 3 + 0.008) ]
```

For example, considering "average_joblog_size_MB = 0.001 MB (1KB)", you obtain:

```
1.000
        jobs_per_day: 370 MB --> volume_size = 370Mi
```

```
10.000
        jobs_per_day: 2.6 GB --> volume_size = 2600Mi
```

```
100.000
        jobs_per_day: 25 GB --> volume_size = 25Gi
```

Deploying Docker compose on Linux on Z

Before you deploy IBM Workload Automation components on Linux on Z, ensure that you have deployed Docker compose, as explained in the following procedure.

To deploy the containers, docker-compose is required on the local workstation. Perform the following steps:

1. Browse to `/usr/local/bin` and create a file with name `docker-compose` with the following contents:

```
#
# This script will attempt to mirror the host paths by using volumes for the
# following paths:
#   * $(pwd)
#   * $(dirname $COMPOSE_FILE) if it's set
#   * $HOME if it's set
#
# You can add additional volumes (or any docker run options) using
# the $COMPOSE_OPTIONS environment variable.
#


set -e

VERSION="1.27.4"
IMAGE="ibmcom/dockercompose-s390x:$VERSION"


# Setup options for connecting to docker host
if [ -z "$DOCKER_HOST" ]; then
    DOCKER_HOST='unix:///var/run/docker.sock'
fi
if [ -S "${DOCKER_HOST#unix://}" ]; then
    DOCKER_ADDR="-v ${DOCKER_HOST#unix://}:${DOCKER_HOST#unix://} -e DOCKER_HOST"
else
    DOCKER_ADDR="-e DOCKER_HOST -e DOCKER_TLS_VERIFY -e DOCKER_CERT_PATH"
fi


# Setup volume mounts for compose config and context
if [ "$(pwd)" != '/' ]; then
    VOLUMES="-v $(pwd):$(pwd)"
fi
if [ -n "$COMPOSE_FILE" ]; then
    COMPOSE_OPTIONS="$COMPOSE_OPTIONS -e COMPOSE_FILE=$COMPOSE_FILE"
    compose_dir="$(dirname "$COMPOSE_FILE")"
    # canonicalize dir, do not use realpath or readlink -f
    # since they are not available in some systems (e.g. macOS).
    compose_dir="$(cd "$compose_dir" && pwd)"
fi
if [ -n "$COMPOSE_PROJECT_NAME" ]; then
    COMPOSE_OPTIONS="-e COMPOSE_PROJECT_NAME $COMPOSE_OPTIONS"
fi
if [ -n "$compose_dir" ]; then
    VOLUMES="$VOLUMES -v $compose_dir:$compose_dir"
fi
if [ -n "$HOME" ]; then
    VOLUMES="$VOLUMES -v $HOME:$HOME -e HOME" # Pass in HOME to share docker.config and allow
 ~/-relative paths to work.
fi
i=$#
while [ $i -gt 0 ]; do
    arg=$1
    i=$((i - 1))
    shift

    case "$arg" in
        -f|--file)
            value=$1
            i=$((i - 1))
            shift
```

```
            set -- "$@" "$arg" "$value"

            file_dir=$(realpath "$(dirname "$value")")
            VOLUMES="$VOLUMES -v $file_dir:$file_dir"
        ;;
        *) set -- "$@" "$arg" ;;
    esac
done

# Setup environment variables for compose config and context
ENV_OPTIONS=$(printenv | sed -E "/^PATH=.*/d; s/^/-e /g; s/=.*//g; s/\n/ /g")

# Only allocate tty if we detect one
if [ -t 0 ] && [ -t 1 ]; then
    DOCKER_RUN_OPTIONS="$DOCKER_RUN_OPTIONS -t"
fi

# Always set -i to support piped and terminal input in run/exec
DOCKER_RUN_OPTIONS="$DOCKER_RUN_OPTIONS -i"


# Handle userns security
if docker info --format '{{json .SecurityOptions}}' 2>/dev/null | grep -q 'name=userns'; then
    DOCKER_RUN_OPTIONS="$DOCKER_RUN_OPTIONS --userns=host"
fi

# shellcheck disable=SC2086
exec docker run --rm $DOCKER_RUN_OPTIONS $DOCKER_ADDR $COMPOSE_OPTIONS $ENV_OPTIONS $VOLUMES -w
 "$(pwd)" $IMAGE "$@"
```

2. Run the following command to make the `docker-compose` file an executable file:

```
sudo chmod +x /usr/local/bin/docker-compose
```

3. More detailed technical information for each component are available in the sample readme files:
   - IBM Workload Automation Server
   - IBM Workload Automation Console
   - IBM Workload Automation dynamic agent
   - IBM Workload Automation z-centric agent

Deploying containers with Docker

How to deploy the current version of IBM Workload Automation using Docker containers.

This chapter describes how to deploy the current version of IBM Workload Automation using Docker containers.

The available Docker containers are:

- IBM Workload Automation Server, containing the master domain manager and backup master domain manager images for UNIX, Windows, and Linux on Z operating systems.
- IBM Workload Automation Console, containing the Dynamic Workload Console image for UNIX, Windows, Linux on Z operating systems, and the IBM z/OS Container Extensions (zCX) feature.
- IBM Workload Automation dynamic agent and the image of the agent with the machine learning engine, containing the Agent image for UNIX, Windows, Linux on Z operating systems. and the IBM z/OS Container Extensions (zCX) feature.
- IBM Workload Automation z-centric agent, containing the Agent image for UNIX, Windows, Linux on Z operating systems. and the IBM z/OS Container Extensions (zCX) feature.

Each container package includes also a `docker-compose.yml` file to configure your installation.

The dynamic agent component (also the one included in the IBM Workload Automation Server container) is deployed and configured with a gateway.

You can choose to deploy all product containers with a single command, or you can deploy each product component container individually.

Deploying all product component containers with a single command

The following readme file contains all the steps required to deploy all product components at the same time: IBM Workload Automation

Deploying each product component container individually

If you want to install server, console and agent containers individually, see the related readme files :

- IBM Workload Automation Server
- IBM Workload Automation Console
- IBM Workload Automation dynamic agent
- IBM Workload Automation z-centric agent

Note: The database is always external to the Docker engine and is not available as a container

Note: When deploying the server (master domain manager) container, the database schema is automatically created at the container start. If you are planning to install both the IBM Workload Automation server master domain manager and backup master domain manager, ensure that you run the command for one component at a time. To avoid database conflicts, start the second component only when the first component has completed successfully.

Accessing the Docker containers

This topic shows you how to access the container shell and run IBM Workload Automation commands.

To check the container status and run IBM Workload Automation commands, you need to access the containers as described below:

1. Obtain the container ID by running the following command: docker ps

   An output similar to the following one is returned:

   ```
   CONTAINER ID        IMAGE        NAMES               ........        .......

   b02459af2b9c        ......       wa-console          ........        .......
   ```

2. Access the Docker container by running the following command: docker exec -it <container_id> /bin/bash

   Where

   container_id

   Is the ID of the container obtained with the command explained in the first step, for example **b02459af2b9c**.

Connecting an on-prem fault tolerant agent to an IBM Workload Automation Server container

To establish a communication between an on-prem fault tolerant agent and an IBM Workload Automation server container, configure the server *docker-compose.yml* file as follows:

1. Open all external ports as shown in the example below:

   ```
   ports (port mapping "external:internal"):
             - 31116:31116 #HTTPS server port
             - 31111:31111 #HTTP netman port
             - 33113:33113 #HTTPS netman ssl port
             - 31131:31131 #HTTP EIF port
             - 35131:35131 #HTTPS EIF ssl port
   ```

2. Add the **extra_hosts** parameter under **hostname**, and insert all remote machine hostnames that docker must contact (including the one of the on-prem FTA).

```
hostname: wa-server
        extra_hosts:
        - hostname1:IP_Address
        - hostname2:IP_Address
        - hostname3:IP_Address
        ...
```

Furthermore, in the `/etc/hosts` file on the remote machine where the on-prem FTA is running, add the hostname of the IBM Workload Automation server container.

```
IP_Address    hostname
```

Note: You can find the hostname of the IBM Workload Automation server container in the server *docker-compose.yml* file.

Creating a Docker image to run dynamic agents

Quickly create a Docker image to run dynamic agents.

You can run dynamic agents in a Docker container that you use to run jobs remotely, for example to call REST APIs or database stored procedures, or to run jobs within the container itself.

To create a Docker container, you are provided with step-by-step instructions and the latest versions of the required samples on GitHub here. Follow the instructions to create a Docker container to run jobs remotely, or use it as base image to add the applications to be run with the agent to other images, or customize the samples to best meet your requirements.

Deploying IBM Workload Automation components on Red Hat OpenShift

You can now deploy IBM Workload Automation components on Red Hat OpenShift.

IBM Workload Automation supports Red Hat OpenShift, 4.*x* for all IBM Workload Automation product components.

Three separate containers are provided:

- IBM Workload Automation agent
- IBM Workload Automation server
- Dynamic Workload Console

You can deploy the IBM Workload Automation components following one of the below procedures:

Deploying IBM Workload Automation components using helm charts

You can deploy the IBM Workload Automation components using IBM® certified containers. For further information, see Deploying IBM Workload Automation components using helm charts *(on page 123)*.

Deploying IBM Workload Automation components using IBM® Workload Automation Operator

You can deploy the IBM® Workload Automation Operator on your Red Hat OpenShift cluster first, and then use the operator to install the IBM Workload Automation components: the IBM Workload Automation server (master domain manager), Dynamic Workload Console, and the dynamic agent. IBM® certified containers are provided for the operator and for each of the product components. For more information, see Deploying IBM Workload Automation components using IBM Workload Automation Operator *(on page 124)*.

Deploying IBM Workload Automation components using helm charts

Deploy the IBM Workload Automation product component containers on a Red Hat OpenShift, V4.*x* environment by using helm charts.

The IBM Workload Automation product components can be deployed onto Red Hat OpenShift, V4.*x*. You can deploy IBM Workload Automation components using IBM® certified containers on a Kubernetes-based container application platform useful

to orchestrate containerized applications. You can then manage the IBM Workload Automation containers from the OpenShift dashboard or from the command line interface.

For technical information about the deployment, see Deploy IBM Workload Automation using helm charts.

Deploying IBM Workload Automation components using IBM® Workload Automation Operator

Deploy the IBM Workload Automation components on Red Hat OpenShift by using IBM® Workload Automation Operator.

For technical information about the deployment of the IBM Workload Automation components on Red Hat OpenShift by using the IBM® Workload Automation Operator, see the following readme files:

- IBM Workload Automation
- Deploying the IBM Workload Automation Operator
- Deploying the IBM Workload Automation components

Deploying on Amazon EKS

You can use Amazon Elastic Kubernetes Service (EKS) to run IBM® Workload Scheduler containers on Amazon Web Services (AWS) EKS.

As more and more organizations move their critical workloads to the cloud, there is an increasing demand for solutions and services that help them easily migrate and manage their cloud environment.

To respond to the growing request to make automation opportunities more accessible, IBM® Workload Scheduler is now offered on the Amazon Web Services cloud. Within just a few minutes, you can access the product Helm chart and container images and easily launch an instance to deploy an IBM® Workload Scheduler server, console, and agents with full on-premises capabilities on AWS. IBM® Workload Scheduler on AWS improves flexibility and scalability of your automation environment. It helps in lowering costs and eliminating complexity, while reducing the operational overhead and the burden involved in managing your own infrastructure, so you can invest your time and resources in growing your business. Also, IBM® Workload Scheduler on AWS delivers faster access to managed services solutions, for a full product lifecycle management.

Full details and deployment instructions are available in the IBM Workload Automation Chart readme file.

Deploying on Azure AKS

You can deploy and manage IBM® Workload Scheduler containers on Azure Kubernetes Service (AKS).

Deploy and manage IBM® Workload Scheduler containerized product components on the Azure AKS, a container orchestration service available on the Microsoft Azure public cloud. You can use Azure AKS to deploy, scale up, scale down and manage containers in the cluster environment. You can also deploy and run an Azure SQL database.

As more and more organizations move their critical workloads to the cloud, there is an increasing demand for solutions and services that help them easily migrate and manage their cloud environment.

To respond to the growing request to make automation opportunities more accessible, IBM® Workload Scheduler can now be deployed on Azure AKS. Within just a few minutes, you can easily launch an instance to deploy an IBM® Workload Scheduler server, console, and agents with full on-premises capabilities on the Microsoft Azure public cloud.

IBM® Workload Scheduler deployed in a cluster environment improves flexibility and scalability of your automation environment. It helps in lowering costs and eliminating complexity, while reducing the operational overhead and the burden involved in managing your own infrastructure, so you can invest your time and resources in growing your business.

Running the product containers within Azure AKS gives you access to services such as a highly scalable cloud database service. You can deploy and run any of the following Azure SQL Server database models in the Azure cloud, depending on your needs:

- SQL database
- SQL managed instance
- SQL virtual machine

Full details and deployment instructions are available in the IBM Workload Automation Chart readme file.

Deploying on Google GKE

You can deploy and manage IBM® Workload Scheduler containers on Google GKE.

Google Kubernetes Engine (GKE) provides a managed environment for deploying, managing, and scaling your containerized applications using Google infrastructure. The Google GKE environment consists of multiple machines grouped together to form a cluster. You can also deploy and run Google Cloud SQL for SQL server.

As more and more organizations move their critical workloads to the cloud, there is an increasing demand for solutions and services that help them easily migrate and manage their cloud environment.

To respond to the growing request to make automation opportunities more accessible, IBM® Workload Scheduler can now be deployed on Google GKE. Within just a few minutes, you can easily launch an instance to deploy an IBM® Workload Scheduler server, console, and agents with full on-premises capabilities on the Google GKE public cloud.

IBM® Workload Scheduler deployed in a cluster environment improves flexibility and scalability of your automation environment. It helps in lowering costs and eliminating complexity, while reducing the operational overhead and the burden involved in managing your own infrastructure, so you can invest your time and resources in growing your business.

Running the product containers within Google GKE gives you access to services, such as a cloud database service. Cloud SQL for SQL Server is a managed database service that helps you set up, maintain, manage, and administer your SQL Server databases on Google Cloud Platform

Full details and deployment instructions are available in the IBM Workload Automation Chart readme file.

Deploying AI Data Advisor

You can deploy AI Data Advisor (AIDA) by using Docker or Kubernetes.

AIDA is composed of two major components: AIDA Exporter and AIDA Engine. Each component contains a number of services:

AIDA Exporter

Exporter

Through IBM Workload Scheduler APIs, exports KPIs metrics from IBM Workload Scheduler (according to OpenMetrics standard) and stores them into AIDA OpenSearch database.

Also, it exports Alert definitions from IBM Workload Scheduler and imports them into OpenSearch.

AIDA Engine

Predictor

Calculates the expected values of each KPI, also considering special days.

Anomaly detection and alert generation

Detects anomalies in KPIs trend by comparing observed KPI data points with expected values, and generates alerts when trigger conditions are met.

Email notification

Sends email notification when alerts are generated.

Orchestrator

Orchestrates KPI prediction and anomaly detection.

UI

AIDA User Interface.

Internal event manager

Manages communication among AIDA services.

Also, AIDA uses:

OpenSearch (an Elasticsearch technology)

To store and analyze data.

Keycloak (optional)

To manage security and user access in AIDA (Docker deployment only). Keycloak is optional. If not deployed, the Dynamic Workload Console user authentication roles will be used.

Nginx

As a reverse proxy for its components.

Deploying AIDA using Docker

To monitor IBM Workload Scheduler and IBM® Z Workload Scheduler engines, you can deploy AIDA with Docker by using AIDA.sh script. This script provides options to run Docker Compose operations and AIDA configuration steps.

For details, see the following readme file for Docker:

- HCL AI Data Advisor for IBM Workload Automation

Deploying AIDA using Kubernetes

To monitor IBM Workload Scheduler engines only (not IBM® Z Workload Scheduler engines), you can deploy AIDA by using an **helm chart**. This helm chart is included in the Workload Automation product helm chart that allows you to deploy Workload Automation and all its components in one shot.

For details, see the following readme file for Kubernetes:

- HCL AI Data Advisor for IBM Workload Automation

Note: Horizontal pod autoscaling based on memory and network usage is supported for **Anomaly detection and alert generation** and **Predictor** services. In case of high memory utilization, Kubernetes replicates pods. When memory usage decreases, pods are deleted.

Troubleshooting

This section describes how to resolve problems with IBM Workload Automation containers. It describes the tools available to help you troubleshoot problems and details many known problem scenarios, and their solutions.

Container maintenance procedure

Check how to avoid POD restart during maintenance.

The POD status check of a Kubernetes-based environment is based on Liveness Probe; the latter checks if all processes are active, if one or more processes are not active, the POD is automatically restarted. Therefore, in case of maintenance, manually stopping the IBM Workload Scheduler processes, the Dynamic Workload Console processes, or the dynamic agent processes causes a POD restart.

To avoid a PD restart during maintenance:

- In the selected POD, create the following file: `/opt/wautils/wa_maintenance`
- Stop the processes listed above and perform all the steps needed for the maintenance
- Restart the stopped processes
- Delete the `/opt/wautils/wa_maintenance` file

Container deployment issues

Check the steps to do if you run into deployment issues.

If a problem occurs during the deployment, check the steps described below to solve it.

Docker compose

1. Check the system requirements Prerequisite information when deploying with containers *(on page 119)*
2. Make sure that all required configuration parameters have been correctly configured (e.g. license, WA_PASSWORD, DB parameters, etc.).
3. Make sure that the external port mapping does not collide with ports already used by other processes.
4. Activate the debug mode by performing the following steps:
    - Remove all containers by launching the "docker rm -f wa-server wa-console wa-db2" command.
    - Remove the associated volumes by launching the "docker volume prune" command.
    - Edit the docker-compose.yml file by adding “- WA_DEBUG=yes? under the environment variables (this prevents the containers to exit after the failure).
    - Launch again the services by using "docker-compose up -d".
    - Enter the container name by using "docker exec -it *container_name_or_id* /bin/bash".
    - Check the logs
5. If you find an error in the logs, check the detailed logs in /home/wauser/wadata/installation/logs

Red Hat OpenShift

1. See the system requirements documented in the readme IBM Workload Automation for OpenShift V4.*x*.
2. Make sure that all required configuration parameters have been correctly configured in the custom resources (OpenShift 4.*x*) or in the *template.yml* file (OpenShift 3.*x*), for example, license, pools, storage class, to name a few.
3. From the OpenShift command line, check the POD logs by launching the "oc logs -f *pod_name*" command. If launched with the -f (--follow) option, it shows useful information about the installation phase. From the OpenShift platform, go to *Stateful Sets* section in *Applications*, double click on PODS and then click on the POD's name to see the related logs.
4. Activate the debug mode to check the container installation and configuration logs by setting true the "WA_DEBUG" parameter in the configuration file.
5. If you find an error in the logs, check the detailed logs in `/home/wauser/wadata/installation/logs`

"CURL error 35" error

This document explains how to solve the *CURL error 35* error that might occur on the agent.

If in the *JobManagerGW_message.log* file on the agent you find the following error:

```
|18446744072657463040|152|agent-95-waagent-0.agent-95-waagent-h.cert-manager.svc.cluster.local|
AWSITA320E The gateway was not able to contact the broker server at the address
"https://localhost:35116/JobManagerRESTWeb/JobSchedulerGW/actions/GWID_AGENT_ICP_agent_95_waagent_0"
to obtain the list of actions to execute.
The error is: "AWSITA245E An error occurred getting the response of the HTTP request.
The error is "CURL error 35".
```

And simultaneously in the *message.log* on the server you find the following error:

```
00058543 com.ibm.scheduling.jobdispatcher
W AWKJDE235W A connection problem occurred submitting job ID "25f769bd-d0e3-3a90-ae47-c7f8a51c549c" with name
"AGENT_ICP#EVERY_1800_4.S_PEAK_JOB_65.SCHEDID-0AAAAAAAAAAP35AZ.JNUM-757735705" to the endpoint URL
"https://agent-95-waagent-0:31114/ita/JobManager/job".
The error message is: "AWKJDE519E The agent did not contact the server to manage this request.".
```

Proceed as follows:

1. Edit the *JobManagerGW.ini* file on the agent, by adding **ActionPollers = 3** (if the ActionPollers is not specified, the default value is 1). The file is located in the following path:

    ```
    /home/wauser/wadata/ITA/cpa/config/
    ```

Note: The **ActionPollers = 3** must be added only in the *[ITA]* section.

The following is an example of the *JobManagerGW.ini* file:

```
[ITA]
name = JobManagerGW
autostart = yes
fname = /opt/wa/TWS/bin/JobManagerGW
keepalive = yes
status_timeout = 300
check_status = yes
commstart = false
display_name = JobManagerGW
version = 1.0
type = optional
min_up_time = 60
JobManagerGWID = GWID_AGENT_ICP_agent_95_waagent_0
JobManagerGWURIs = https://localhost:31114/ita/JobManagerGW/JobManagerRESTWeb/
                   JobScheduler/resource
ActionPollers = 3
```

2. To avoid a POD restart during maintenance, follow the procedure described in Container maintenance procedure *(on page 126)*.
3. Stop and start the agent by submitting the following command:

   /opt/wa/TWS/ShutDownLwa

   /opt/wa/TWS/StartUpLwa

Chapter 3. Post-installation configuration

The most common configuration steps to be performed after completing the installation.

After successfully installing IBM Workload Scheduler, there are a number of recommended configuration steps to be performed that are described in more detail in this section. Also consider the FAQs listed below.

FAQ - Security configurations

A list of questions and answers related to security configurations:

When installing the IBM Workload Scheduler, you might have the need to customize some parameters to suit your environment.

Can I install any IBM Workload Scheduler components without setting up the SSL configuration?

No, starting from version 10.2.1, certificates, either default or custom, are required when installing IBM® Workload Scheduler. You can no longer install IBM® Workload Scheduler without securing your environment with certificates.

Q:How do I set up SSL communication using custom certificates for master domain manager and dynamic agent?

See the detailed explanation in Customizing certificates for master domain manager and dynamic agent communication *(on page)*.

Q:How do I set up SSL communication using custom certificates for master domain manager and Dynamic Workload Console?

See the detailed explanation in Customizing certificates for master domain manager and Dynamic Workload Console communication *(on page)*.

Q: How do I set up SSL communication using custom certificates for agents?

You can use the **-sslkeysfolder** and **--sslpassword** parameters when installing fault-tolerant agents and dynamic agents with the twsinst command. For more information, see Agent installation parameters - twsinst script *(on page 84)*.

How do I configure master domain manager and dynamic domain manager in SSL mode?

See the detailed explanation in Configuring your master domain manager and dynamic domain manager in SSL mode *(on page 136)*.

Q: Is FIPS supported?

FIPS compliance is not supported in the current product version. This is because OpenSSL 3.0 libraries do not provide a FIPS-compliant validation algorithm for P12 certificates. Development is in progress with the aim of supporting FIPS in upcoming releases.

How do I configure Single Sign-On?

Single Sign-On (SSO) is a method of access control that allows a user to authenticate once and gain access to the resources of multiple applications sharing the same user registry. For more information, see Configuring the Dynamic Workload Console for Single Sign-On *(on page)*.

Configuring a user registry

In this topic you can find information about how to configure a user registry.

By default, the dynamic domain manager, the Dynamic Workload Console, and the master domain manager are configured to use a local file-based user repository. For more information about supported authentication mechanisms, see Available configurations *(on page)*.

You can implement an OpenID Connect (OIDC) user registry, a Lightweight Directory Access Protocol (LDAP) user registry, or a basic user registry by configuring the sample authentication templates provided in XML format. You can further customize the templates by adding additional elements to the XML files. For a full list of the elements that you can configure to complement or modify the configuration, see the related WebSphere Application Server Liberty Base documentation, for example LDAP User Registry (ldapRegistry).

To configure an OIDC user registry, see Configuring an OIDC user registry *(on page 130)* .

To configure an LDAP user registry, for example as Active Directory, see Configuring an LDAP user registry *(on page 131)*.

To configure a basic user registry, see Configuring a basic user registry *(on page 132)*.

Configuring an OIDC user registry

You can implement an OIDC user registry by configuring the sample authentication template provided in XML format: `openid_connect.xml`.

To configure an OIDC user registry, complete the following steps:

1. Copy the following template to a working directory:

```
        <server>
              <featureManager>
                <feature>openidConnectClient-1.0</feature>
              </featureManager>
              <authFilter id="restFilterOpenID">
              <requestUrl id="restUrl"
 urlPattern="jwt/ibm/api|/dwc/rest/roles|/dwc/ServiceDispatcherServlet?ServiceName=PrefExport|/metric
s|/dwc/AjaxServiceDispatcherServlet" matchType="notContain"/>
              </authFilter>
               <openidConnectClient id="keycloak"
                    clientId="wa-service"
                    clientSecret="put_oidc_secret_here"
                    httpsRequired="true"
                    userIdentifier="preferred_username"
                    signatureAlgorithm="RS256"
                    scope="openid"
                    authFilterRef="restFilterOpenID"
                    inboundPropagation="supported"
                    groupIdentifier="groups"
                    redirectToRPHostAndPort="https://dwc_ingress_hostname"
                    discoveryEndpointUrl="https://oidc ingress hostname/realms/wa
                    /.well-known/openid-configuration
                    hostNameVerificationEnabled="false">"
                </openidConnectClient>
        </server>
```

2. Edit the template file in the working folder with the desired configuration by adding users and groups as necessary.
3. Optionally, create a backup copy of the configuration file in the `overrides` folder, if already present.
4. Copy the updated template file to the `overrides` folder.
5. To upload the certificates of the OIDC provider, browse to `<DWC_home>/java/jre/bin` and run the following command:

```
keytool -importcert -file ingress-cert.pem
 -keystore <DWC_home>/usr/servers/dwcServer/resources/security/TWSServerTrustFile.p12 -alias
 ingress-cert -storepass <password_keystore>
```

   where

   ingress-cert.pem

   The certificates file to be imported into the Dynamic Workload Console.

   ingress-cert

   The alias defined during the import of the certificate.

Note: If one or more messages similar to the following are displayed, perform the steps listed below.

```
CWPKI0819I: The default keystore is not created because a password is not configured on the
<keyStore id="defaultKeyStore"/> element, and the 'keystore_password' environment variable is not set.
CWOAU0073E: An authentication error occurred. Try closing the web browser and authenticating again,
or contact the site administrator if the problem persists.
```

1. On the workstation where the Dynamic Workload Console is installed, browse to the `server.xml` file located in `<dwc_installation_directory>/usr/servers/dwcServer`.
2. Open the file with a text editor and change the value of the **sameSiteCookie** parameter from `Strict` to `lax`.
3. Optionally, trust the Dynamic Workload Console certificate with the keycloak certificate.

Configuring an LDAP user registry

You can implement an LDAP based user repository by configuring the following sample authentication templates provided in XML format. The following are the supported authentication methods and the corresponding sample template that can be configured to replace the configuration file currently in use:

- OpenLDAP: `auth_OpenLDAP_config.xml`
- IBM® Directory Server: `auth_IDS_config.xml`
- Windows Server Active Directory: `auth_AD_config.xml`

To configure a common authentication provider for both the IBM® Workload Scheduler and the Dynamic Workload Console, complete the following steps:

1. Assign a role to your authentication provider user or group.
    a. Log in to the Dynamic Workload Console as administrator and access the **Manage Roles** page.
    b. Add a new **Entity** of type **Group** to the role you want to assign to your authentication provider user or group and click **Save**.
2. Update the authentication configuration template file with the details about your authentication provider server.

    a. Copy the template file to a working directory. The templates are located in the following path:

    Dynamic Workload Console

    ```
    DWC_DATA_dir/usr/servers/dwcServer/configDropins/templates/authentication
    ```

    master domain manager

    ```
    TWA_DATA_DIR/usr/servers/engineServer/configDropins/templates/authentication
    ```

    Dynamic Workload Console

    ```
    DWC_home\usr\servers\dwcServer\configDropins\templates\authentication
    ```

    master domain manager

    ```
    TWA_home\usr\servers\engineServer\configDropins\templates\authentication
    ```

    b. Edit the template file in the working directory with the desired configuration.

    c. Optionally, create a backup copy of the configuration file in a different directory, if the file is already present. To avoid conflicts, ensure the backup copy is in a directory different from the following directories: `configDropins/templates` and `configDropins/overrides`.

    d. Copy the updated template file to the `overrides` directory.

    e. The `overrides` directory is located in the following path:

    Dynamic Workload Console

    ```
    DWC_DATA_dir/usr/servers/dwcServer/configDropins/overrides
    ```

    master domain manager

```
TWA_DATA_DIR/usr/servers/engineServer/configDropins/overrides
```

Dynamic Workload Console

```
DWC_home\usr\servers\dwcServer\configDropins\overrides
```

master domain manager

```
TWA_home\usr\servers\engineServer\configDropins\overrides
```

f. Stop and restart WebSphere Application Server Liberty Base using the stopappserver and startappserver commands located in `TWA_home/appservertools`.

For more information about configuring an LDAP registry, see the WebSphere Application Server Liberty Base documentation, for example: Configuring LDAP user registries in Liberty and Federation of user registries.

Configuring a basic user registry

You might want to use a basic user registry by defining the users and groups information for authentication on WebSphere Application Server Liberty Base, even though this type of authentication is not recommended. This type of authentication cannot be used for production, but only for test purposes.

To configure basic user registry, complete the following steps:

1. Copy the `auth_basicRegistry_config.xml` template from the `templates` folder to a working folder.
2. Edit the template file in the working folder with the desired configuration by adding users and groups as necessary.

   To add a user, add an entry similar to the following in the **basicRegistry** section:

   ```
   <user name="nonadminuser" password="{xor}Ozo5PiozKw==">
   ```

   To add a group, add an entry similar to the following in the **basicRegistry** section:

   ```
   <group name="TWSUsers">
          <member name="nonadminuser"/>
          </group>
   ```

3. Store the password in xor, aes, or hash formats using the WebSphere Application Server Liberty Base securityUtility command, as described in securityUtility command.

   This utility requires the JAVA_HOME environment variable to be set. If you do not have Java installed, you can optionally use the Java version provided with the product and available in:

   IBM® Workload Scheduler

   `<INST_DIR>/TWS/JavaExt/jre/jre`

   Dynamic Workload Console

   `<DWC_INST_DIR>/java/jre/bin`

4. Create a backup copy of the configuration file in the `overrides` folder, if already present.
5. Copy the updated template file to the `overrides` folder. Maintaining the original folder structure is not required.

WebSphere Application Server Liberty Base configuration

Describes how WebSphere Application Server Liberty Base configuration files are organized in IBM Workload Scheduler

To simplify administration, configuration, and backup and recovery on UNIX systems, a new default behavior has been implemented with regard to the storage of product data and data generated by IBM Workload Scheduler, such as logs and configuration information. These files are now stored by default in the *TWA_DATA_DIR* directory, which you can optionally customize at installation time.

With a similar approach, also the configuration files for WebSphere Application Server Liberty Base on UNIX systems are stored in the *TWA_DATA_DIR* directory, while binary files are stored in TWA_home.

Also, configuration settings, usually stored in the `server.xml` file, are now divided into several `.xml` files.

To modify WebSphere Application Server Liberty Base configuration settings, first find out the `.xml` file to be modified and the directory where it is stored.

Table 12: WebSphere Application Server Liberty Base configuration files *(on page 133)* lists the files available for WebSphere Application Server Liberty Base configuration.

Table 12. WebSphere Application Server Liberty Base configuration files

Configuration files for IBM Workload Scheduler and WebSphere Application Server Liberty Base

Configuration file	Functionality
TWA_DATA_DIR`/usr/servers/engineServer/configDropins/overrides`	
`authentication_config.xml`	Authentication settings
`datasource.xml`	Datasource settings
`host_variables.xml`	Hostname and port settings
`jvm.options`	Settings for Java Virtual machine, such as **HeapSize**
`ports_variables.xml`	Hostname and port settings
`ssl_variables.xml`	SSL connections and certificates
`wauser_variables.xml`	Authentication settings
TWA_DATA_DIR`/usr/servers/engineServer/resources/security`	
`TWSServerKeyFile.p12`	WebSphere Application Server Liberty Base key store file, containing security keys
`TWSServerTrustFile.p12`	WebSphere Application Server Liberty Base trust store file, containing certificates
`ltpa.keys`	LTPA keys, to be configured for Single Sign On

On Windows systems, there is no such separation and the path to WebSphere Application Server Liberty Base configuration files is as follows:

On master domain managers

<*TWA_home*>`\usr\servers\engineServer\configDropins\overrides`

On Dynamic Workload Console

<*DWC_home*>`\usr\servers\dwcServer\configDropins\overrides`

For more information about using templates to configure WebSphere Application Server Liberty Base to work with IBM Workload Scheduler, see Configuring IBM Workload Scheduler using templates *(on page)*.

Configuring the TLS V1.3 security protocol

The following procedures enable you to configure the TLS V1.3 security protocol for IBM Workload Scheduler. If you want to configure your environment with the TLS V1.3 protocol, it is recommended to use a 4k-length certificate.

Note: TLS V1.3 security protocol support is available from IBM® Workload Scheduler version 10.1 FP4 onwards.

The configuration of the TLS V1.3 security protocol can be manually done on every component:

The configuration of the TLS V1.3 security protocol can only be set using custom certificates with RSA keys of at least *2K*.

Dynamic agents

To enable the TLS V1.3 security protocol for dynamic agents you must open the `<TWSDATA>/ITA/cpa/ita/ita.ini` file and go to the *ITA SSL* section. Here you can set the security modifying the following keywords:

Enabling the TLS V1.3 security protocol exclusively

```
ssl version= TLSv1.3
ssl_ciphers=
```

Enabling the TLS V1.2 and TLS V1.3 security protocols

```
ssl version= atleast.TLSv1.2
ssl_ciphers=
```

where:

ssl_version

Specify the SSL version to be used. Supported values are:

- **atleast.TLSv1.0**
- **atleast.TLSv1.1**
- **atleast.TLSv1.2**
- **atleast.TLSv1.3**

where you specify the minimum version of the TLS protocol to be used. In this case, IBM® Workload Scheduler uses the specified version of the protocol or a higher version, if supported.

- **max.TLSv1.0**
- **max.TLSv1.1**
- **max.TLSv1.2**
- **max.TLSv1.3**

where you specify the maximum version of the TLS protocol to be used. In this case, IBM® Workload Scheduler uses the specified version of the protocol or a lower version.

- **TLSv1.0**
- **TLSv1.1**
- **TLSv1.2**
- **TLSv1.3**

where you specify the exact version of the TLS protocol to be used. In this case, IBM® Workload Scheduler uses the specified version of the protocol.

ssl_ciphers

Define the ciphers that the workstation supports during an SSL connection.

If you want to use an OpenSSL cipher class, use the following command to find out the list of available classes:

```
openssl ciphers
```

For a full list of supported ciphers, see SSL Ciphers and OpenSSL.

Note: The dynamic agents must be restarted after the modifications are completed.

WebSphere Application Server Liberty Base

The following procedures must be repeated for every IBM Workload Scheduler component in the environment that has WebSphere Application Server Liberty Base installed.

To enable the TLS V1.3 security protocol for WebSphere Application Server Liberty Base you must copy the `<TWA_INSTALL_FOLDER>/usr/servers/engineServer/configDropins/defaults/ssl_config.xml` file and paste it in the following folders:

- `<TWA_INSTALL_FOLDER>/usr/servers/engineServer/configDropins/overrides`
- `<DWC_INSTALL_FOLDER>/usr/servers/dwcServer/configDropins/overrides`

You must then edit the `ssl_config.xml` file:

Enabling the TLS V1.3 security protocol exclusively

```
sslProtocol="TLSv1.3"
```

Enabling the TLS V1.2 and TLS V1.3 security protocols

```
sslProtocol="TLSv1.2,TLSv1.3"
```

No spaces can be used before or after the comma.

Note: WebSphere Application Server Liberty Base must be restarted after the modifications are completed.

Native components and fault-tolerant agents

The following procedures must be repeated for every native component and fault-tolerant agents in the IBM Workload Scheduler environment.

To enable the TLS V1.3 security protocol for native components and fault-tolerant agents, you must open the `<TWSDATA>/localopts` file. Choose the procedure that applies to the kind of certificates you are using:

Opens SSL

Enabling the TLS V1.3 security protocol exclusively

Set the **ssl version** keyword as follows:

```
ssl version = TLSv1.3
```

Note: The native components and fault-tolerant agents must be restarted after the modifications are completed.

Using SSL for event-driven workload automation (EDWA) behind firewalls

This feature allows a domain manager to be run as a reverse proxy for HyperText Transfer Protocol (HTTP) and Event Integration Facility (EIF) protocols, forwarding traffic to the Event Processor. An option, enabled using the **optman** command-line program, allows you to choose if workstations that are behind a firewall must connect to the domain manager instead of to the event processor, causing the new proxy on the domain manager to forward its traffic to the event processor.

Restriction: This configuration is not supported if the agent workstation is a dynamic agent.

The incoming traffic is rerouted as follows:

- If an agent is behind a firewall, the traffic is routed to the domain manager on the agent. If an agent is not behind a firewall, the traffic is sent directly to the event processor.
- If domain managers have child nodes behind a firewall, the traffic is rerouted to the event processor.

- Primary domain managers always reroute traffic to the current event processor.
- Lower level domain managers reroute traffic to upper level domain managers if they are behind a firewall, or to the event processor if they are not behind a firewall.

To use this feature, perform the following steps:

1. Enable the feature by setting the **optman** option to `yes`. The default value is `no`:
   ```
   enEventDrivenWorkloadAutomationProxy | pr = {yes|no}
   ```
2. In the workstation definition in the database for the agent, set the **`behindfirewall`** attribute to `ON`.
3. Configure OpenSSL on the domain manager.

For details about setting the **`behindfirewall`** attribute, see Workstation definition *(on page)*.

Configuring your master domain manager and dynamic domain manager in SSL mode

Configuring your master domain manager and dynamic domain manager in SSL mode

By default, starting from version 10.1 master domain manager and dynamic domain manager are installed in SSL mode.

If you are upgrading from a version earlier than 10.1 and want to set up your master domain manager and dynamic domain manager in SSL mode, perform the following steps:

1. Install the master domain manager or upgrade your current master domain manager to the latest version, for example version 10.2.
2. Stop WebSphere Application Server Liberty Base, as described in Application server - starting and stopping *(on page)*.
3. Replace the values of the following parameters in the `localopts` file with the following values:
   - **nm SSL full port** = `31113`
   - **SSL key** =`TWA_home/TWS/ssl/OpenSSL/TWSClient.key`
   - **SSL certificate** = `TWA_home/TWS/ssl/OpenSSL/TWSClient.cer`
   - **SSL key pwd** = `TWA_home/TWS/ssl/OpenSSL/password.sth`
   - **SSL CA certificate** = `TWA_home/TWS/ssl/OpenSSL/TWSTrustCertificates.cer`
   - **SSL random seed** =`TWA_home/TWS/ssl/OpenSSL/TWS.rnd`
   - **SSL Encryption Cipher** = `TLSv1.2`

   For more information about the `localopts` file, see Setting local options
4. Modify the master domain manager and dynamic domain manager using the composer mod command, as follows:
   ```
   CCPUNAME your_master_domain_manager_workstation

     DESCRIPTION "MANAGER CPU"

     OS UNIX

     NODE localhost TCPADDR 31111

     SECUREADDR 31113

     DOMAIN MASTERDM

     FOR MAESTRO

       TYPE MANAGER

       AUTOLINK ON

       BEHINDFIREWALL OFF

       SECURITYLEVEL FORCE_ENABLED

       FULLSTATUS ON

   END
   ```

```
CPUNAME your_broker_workstation
  DESCRIPTION "This workstation was automatically created."
  OS OTHER
  NODE localhost TCPADDR 41114
  SECUREADDR 41114
  DOMAIN MASTERDM
  FOR MAESTRO
    TYPE BROKER
    AUTOLINK ON
    BEHINDFIREWALL OFF
    SECURITYLEVEL FORCE_ENABLED
    FULLSTATUS OFF
END
```

5. Modify the **Broker.Workstation.PortSSL** parameter in the `BrokerWorkstation.properties` file from `false` to `true`.

   The **Broker.Workstation.PortSSL** parameter specifies the port used by the broker server to listen to the incoming traffic (equivalent to the Netman port) in SSL mode. It is first assigned at installation time. This port number must always be the same for all the broker servers that you define in your IBM® Workload Scheduler network (one with the master domain manager and one with every backup master domain manager you install) to ensure consistency when you switch masters.

6. Start WebSphere Application Server Liberty Base, as described in Application server - starting and stopping *(on page)*.
7. Stop and start all IBM® Workload Scheduler processes.
8. Run

   ```
   Jnextplan -for 0000
   ```

Part III. Configuring

Configuring IBM Workload Scheduler components after installation.

You must configure IBM Workload Scheduler components after installation.

Setting the environment variables

Before you configure your IBM Workload Scheduler components, you must set the environment variables using the `twa_env` or `tws_env` script. You can use the two scripts interchangeably.

The `twa_env` script is located in the following paths:

On Windows operating systems

IBM`/TWA`

On UNIX operating systems

`/opt/`IBM`/TWA`

The upgrade installation process for agents installs a new version of the `tws_env` script in the directory `<`*`TWA_HOME`*`/TWS`, where <*TWA_HOME*> is the IBM Workload Scheduler installation directory. A backup copy of your original version is created in a backup directory. After the upgrade process, merge the content of the new version with the content of the original version to carry your customized content into the new version.

The script is copied into the backup instance in `/<`*`working_dir`*`>/TWA_<`*`user_name_of_installation_user`*`>`

On Windows™ operating systems, run the tws_env.cmd shell script to set up both the *PATH* and *TWS_TISDIR* variables. For example, if IBM Workload Scheduler is installed in the `%ProgramFiles%\IBM\TWA\TWS` directory, the <*PATH*> variable is set as follows:

```
c:\Program Files\IBM\TWA\TWS;c:\Program Files\IBM\TWA\TWS\bin
```

Note: If you have more than one version of IBM Workload Scheduler installed on your computer, make sure <*TWS_TISDIR*> points to the latest one. This ensures that the most recent character set conversion tables are used.

On UNIX™ and Linux™ operating systems, source the . ./tws_env.sh shell script to set up the `PATH`, `TWS_TISDIR`, and `UNISONWORK` variables. For example, if IBM Workload Scheduler is installed in the default directory `/opt/IBM/TWA/TWS` directory, . ./tws_env.sh sets the variables as follows:

```
PATH=/opt/IBM/TWA/TWS:/opt/IBM/TWA/TWS/bin:$PATH
      export PATH

TWS_TISDIR=/opt//opt/IBM/TWA/TWS
     export TWS_TISDIR
```

The tws_env script has two versions:

- tws_env.sh for Bourne and Korn shell environments
- tws_env.csh for C Shell environments

Chapter 1. Configuring a master domain manager

After you installed a master domain manager, follow the steps in this section to add the *FINAL* and *FINALPOSTREPORTS* job streams to the database.

The *FINAL* job stream is placed in production every day and runs JnextPlan before the start of a new day.

The *FINALPOSTREPORTS* job stream, responsible for printing post production reports, follows the *FINAL* job stream and starts only when the last job listed in the *FINAL* job stream (*SWITCHPLAN*) is completed successfully.

The installation creates the `<TWS_INST_DIR>\TWS\Sfinal` file that contains the *FINAL* and *FINALPOSTREPORTS* job stream definitions.

You can use the `<TWS_INST_DIR>\TWS\Sfinal` or create a customized new file for the *FINAL* job stream. For more information, see Customizing and submitting the optional FINAL job stream *(on page 254)* .

The following steps give an example of how to configure a master domain manager after the installation:

1. Log in as *TWS_user* or as administrator.
2. Set the environment variables. See Setting the environment variables *(on page 138)*.
3. Add the *FINAL* and *FINALPOSTREPORTS* job stream definitions to the database by running the following command from the `/opt/IBM/TWA/TWS` directory:

   ```
   composer add Sfinal
   ```

   where *Sfinal* is the name of the file that contains the *FINAL* and *FINALPOSTREPORTS* job stream definitions.
4. Add the *FINAL* and the *FINALPOSTREPORTS* job streams to the plan by running:

   ```
   JnextPlan
   ```

   You can automate this step after installation. See Automating production plan processing *(on page)* .
5. When JnextPlan completes, check the status of IBM Workload Scheduler:

   ```
   conman status
   ```

   If IBM Workload Scheduler started correctly, the status that is returned by the command is `Batchman LIVES`.
6. Change the workstation limit value to run jobs. The default job limit after installation is **0**, so no jobs run at any time. Raise the job limit to allow jobs to run, for example, to run 10 jobs at the same time:

   ```
   conman "limit ;10"
   ```

   If no workstation name is specified for the **limit** command, the default value is the current login workstation.

   Note: If the priority of jobs is **HI** (100) or **GO** (101), the limit is ignored and the jobs run even if the limit is 0, unless the workstation fence is greater than or equal to the priority.

Additionally, the following configuration procedures might be necessary:

- Customizing and configuring global, local, and user options. See the relevant sections in Customizing and configuring IBM Workload Scheduler *(on page)*
- Customizing and configuring user authentication to allow users authorization on actions and objects, and to configure LDAP. See Configuring authentication *(on page)*.
- Setting connection security to enable SSL for inter-component communications. See the relevant sections in Connection security overview *(on page)*.

Chapter 2. Configuring a master domain manager configured as backup

After you install a master domain manager configured as backup, perform the following additional configuration steps:

1. Log in as *TWS_user* on your master domain manager.
2. Add the username and password for the master domain manager configured as backup to the `useropts` file. For details, see Setting user options *(on page)*.
3. Set the environment variables by running tws_env as described in Setting the environment variables *(on page 138)*.
4. Define the master domain manager configured as backup as a full status autolink fault-tolerant agent in the IBM Workload Scheduler database, using the composer command interface or the Dynamic Workload Console. In this example with composer, type the following command:

   ```
   composer
   new
   ```

5. Type the workstation definition in the text editor, for example:

   ```
   CPUNAME BDM1
    DESCRIPTION "Backup master domain mananger"
    OS UNIX
    NODE lab777
    TCPADDR 31111
    FOR MAESTRO
      TYPE FTA
      AUTOLINK ON
      BEHINDFIREWALL OFF
      FULLSTATUS ON
   end
   ```

   For more information about workstation definitions, see Workstation definition *(on page)* .
6. Run JnextPlan -for 0000 to include the master domain manager configured as backup workstation in the plan and to send the Symphony™ file to it.

   Note: Ensure that the global option carryforward is set to all, otherwise only incomplete job streams are carried forward.

7. Change the workstation limit to allow jobs to run on the workstation. For example, set the number of jobs to run concurrently on the workstation to `10`:

   ```
   conman "limit DM1;10"
   ```

   Note: If you are logged into the master domain manager configured as backup, the workstation name (`DM1` in the above example) is not required.

Additionally, the following configuration procedures might be necessary:

- Customizing and configuring global, local, and user options. See the relevant sections in Customizing and configuring IBM Workload Scheduler *(on page)*
- Customizing and configuring user authentication to allow users authorization on actions and objects, and to configure LDAP. See Configuring authentication *(on page)*.
- Setting connection security to enable SSL for inter-component communications. See the relevant sections in Connection security overview *(on page)*.

Chapter 3. Configuring a domain manager

After you install a domain manager, perform the following configuration steps:

1. Log in as *TWS_user* on your master domain manager.
2. Set the environment variables by running tws_env as described in Setting the environment variables *(on page 138)*.
3. Create a new domain by running the following command:

   ```
   composer new domain
   ```

4. Type the domain definition in the text editor, for example:

   ```
   DOMAIN DOMAIN1
     DESCRIPTION "Sample Domain"
     PARENT MASTERDM
   END
   ```

5. Define the domain manager as a full status autolink fault-tolerant agent in the IBM Workload Scheduler database, using the composer command interface or the Dynamic Workload Console. In this example, using composer, type:

   ```
   composer
   new
   ```

6. Type the workstation definition in the text editor, for example:

   ```
   CPUNAME DDM1
     DESCRIPTION "domain mananger"
     OS UNIX
     NODE lab0777
     TCPADDR 31111
     DOMAIN MDM
     FOR MAESTRO
        TYPE MANAGER
        AUTOLINK ON
        BEHINDFIREWALL OFF
        FULLSTATUS ON
   END
   ```

   For more information about workstation definitions, see Workstation definition *(on page)* .
7. Run **JnextPlan -for 0000** to include the domain manager workstation in the plan and to send the Symphony file to it.

   Note: Ensure that the global option carryforward is set to all, otherwise only incomplete job streams are carried forward.

8. Change the workstation limit to allow jobs to run on the workstation. For example, set the number of jobs to run concurrently on the workstation to 10:

   ```
   conman "limit;10"
   ```

Chapter 4. Configuring a backup domain manager

After you install a backup domain manager, perform the following configuration steps:

1. Log in as *TWS_user* on your master domain manager.
2. Set the environment variables by running tws_env as described in Setting the environment variables *(on page 138)*.
3. Define the backup domain manager as a full status autolink fault-tolerant agent in the IBM Workload Scheduler database, using the composer command interface or the Dynamic Workload Console. In this example, using composer, type:

   ```
   composer new
   ```

4. Type the workstation definition in the text editor, for example:

   ```
   CPUNAME Backup_DM
     DESCRIPTION "backup domain mananger"
     OS UNIX
     NODE lab0777
     TCPADDR 31111
     DOMAIN MDM
     FOR MAESTRO
        TYPE FTA
        AUTOLINK ON
        BEHINDFIREWALL OFF
        FULLSTATUS ON
   END
   ```

   For more information about workstation definitions, see Workstation definition *(on page)*.
5. Run **JnextPlan -for 0000** to include the backup domain manager workstation in the plan and to send the Symphony file to it.

   Note: Ensure that the global option carryforward is set to all, otherwise only incomplete job streams are carried forward.

6. Change the workstation limit to allow jobs to run on the workstation. For example, set the number of jobs to run concurrently on the workstation to 10:

   ```
   conman "limit;10"
   ```

Chapter 5. Configuring a dynamic domain manager

After you install a dynamic domain manager, perform the following configuration steps:

1. Log in as *TWS_user* on your master domain manager.
2. Set the environment variables by running tws_env as described in Setting the environment variables *(on page 138)*.
3. Run **JnextPlan -for 0000** to include the dynamic domain manager workstation in the plan and to send the Symphony file to it.

   Note: Ensure that the global option carryforward is set to all, otherwise only incomplete job streams are carried forward.

4. Change the workstation limit to allow jobs to run on the workstation. For example, set the number of jobs to run concurrently on the workstation to 10:

   ```
   conman "limit;10"
   ```

Chapter 6. Configuration steps for a dynamic domain manager configured as backup

After you install a dynamic domain manager as backup, perform the following configuration steps:

1. Log in as *TWS_user* on your master domain manager
2. Set the environment variables by running tws_env as described in Setting the environment variables *(on page 138)*.
3. Define the dynamic domain manager as backup as a full status autolink fault-tolerant agent in the IBM Workload Scheduler database, using the composer command interface or the Dynamic Workload Console. In this example using composer, type:

   ```
   composer
   new
   ```

4. Type the workstation definition in the text editor, for example:

   ```
   CPUNAME BDDM1
     DESCRIPTION "backup dynamic domain mananger"
     OS UNIX
     NODE lab00777
     TCPADDR 31111
     DOMAIN DYNAMICDM
     FOR MAESTRO
        TYPE FTA
        AUTOLINK ON
        BEHINDFIREWALL OFF
        FULLSTATUS ON
   END
   ```

   For more information about workstation definitions, see Workstation definition *(on page)* .
5. Run **JnextPlan -for 0000** to include the dynamic domain manager as backup workstation in the plan and to send the Symphony file to it.

   Note: Ensure that the global option carryforward is set to all, otherwise only incomplete job streams are carried forward.

6. Change the workstation limit to allow jobs to run on the workstation. For example, set the number of jobs to run concurrently on the workstation to 10:

   ```
   conman "limit;10"
   ```

Chapter 7. Configuring a dynamic agent

How to configure a dynamic agent.

The dynamic agent installation process automatically adds the workstation definition to the database and registers the workstation definition to the dynamic workload broker installed on the master domain manager or dynamic domain manager that you chose during the installation process.

Dynamic agents can be organized in pools to help organize your environment based on the availability of workstations and on the requirements of the jobs that need to be run. You can create a pool, adding dynamic agents to a workstation definition of type pool, or, you can automatically register agents to pools through a different process. See Automatically register agents to pools *(on page 145)* for more details.

After installing a dynamic agent, depending on the `enAddWorkstation` global option settings in the master domain manager, perform the following steps:

If `enAddWorkstation` is set to `no`:

1. Run JnextPlan with the **-for 0000** option to add the dynamic agent workstation definition to the plan and to send the Symphony file to it. For more information about workstation definitions, see Workstation definition *(on page)* .

   Note: To carry forward completed and incomplete job stream instances, ensure that the `carryforward` global option is set to `all` or run JnextPlan -for 0000 with the **-noremove** option.

2. Change the workstation limit to allow jobs to run on the workstation. For example, set the number of jobs that can run concurrently on the workstation to 10:

   ```
   conman "limit DA235007_00;10"
   ```

If `enAddWorkstation` is set to `yes`:

The workstation definition is automatically added to the plan after it is defined in the database by the installation process. The `workstationLimit` global option specifies the dynamic agent workstation limit value that the dynamic agent workstation assumes after the workstation is added to the plan.

For more information about how to modify the `enAddWorkstation` and `workstationLimit` global option settings, see Global options - detailed description *(on page)*.

For more information about troubleshooting, see Troubleshooting when automatically adding dynamic agent workstations to the plan *(on page)*.

You might also need to run the following configuration procedures:

- Customizing and configuring `jobmanager.ini` and user options. See Configuring the agent *(on page)*.
- Customizing and configuring `JobManagerGW.ini` for opening communication between the gateway and the dynamic workload broker. See Configuring the agent *(on page)*.
- Customizing and configuring user authentication to allow users authorization on actions and objects, and to configure LDAP. See Configuring authentication *(on page)*.
- Setting connection security to enable SSL for inter-component communications. See the relevant sections in Connection security overview *(on page)*.

Automatically register agents to pools

The dynamic agent installation process automatically adds the workstation definition to the database and registers the workstation definition to the dynamic workload broker installed on the master domain manager or the dynamic domain manager that you specify during the installation process.

You can add dynamic agents in pools to help organize your environment based on the availability of workstations and the requirements of the jobs to be run. Normally, when you create a pool, you add the dynamic agents to a workstation definition of type pool.

Starting from IBM Workload Scheduler version 9.4 Fix Pack 4, you can automatically register dynamic agents in pools by editing the `pools.properties` file located in *TWS_home*`>/ITA/cpa/config`.

Starting from version 9.5, the `pools.properties` file is located in the following paths:

On Windows operating systems

<*TWS_home*>\ITA\cpa\config

On UNIX operating systems

<*TWA_DATA_DIR*>/ITA/cpa/config

This alternative way of registering dynamic agents to a pool can be useful when you need to quickly add more than one agent to a pool, or when you want to associate multiple pools to a dynamic agent.

The file is composed by a series of lines with a list of pools to which the agent will be automatically registered. To make the changes in this file effective on the agent, you must stop the agent, edit the file, then start the agent. See ShutDownLwa - Stop the agent *(on page)* and StartUpLwa - Start the agent *(on page)*.

For example, if you want to register a dynamic agent with three different pools, then edit the `pools.properties` file as follows:

```
POOL1
POOL2
POOL3
```

By default, master domain manager and backup domain manager dynamic agents register with the pool named `MASTERAGENTS`. In this case, the `pools.properties` file on these agents contains the following default entry:

```
$MASTERAGENTS
```

Note: The default name for this pool workstation, MASTERAGENTS, can be modified using the optman global option resubmitJobName. See Global options - detailed description *(on page)* for details about this option.

The following options are supported for each entry in the `pool.properties` file:

;skip

Use this option to exclude pools from even being considered. You might want to ignore specific pools for a period of time, but still maintain them in the list so that they can be considered in the future.

;optional

Use this option to specify that a pool is not obligatory, but optional, so that if the agent is unable to register to a pool, for example, a pool no longer exists) then the pool is ignored.

If an agent has obligatory pools in the `pools.properties` file that are not defined in the system, then the agent will not be able to automatically register and go online. To ensure agent connectivity, these options can be used to manage situations where the agent needs to online even if some pools are not defined.

If the agent does not receive any errors, then the agent goes online and is added to all of the pools in the list, except for those with the ;skip option specified.

If, instead, the agent encounters an error, the agent is able to determine which of the pools in the list has a problem. If the problematic pool is mandatory (without the ;optional option specified), then the agent goes offline and is not added to any of the pools. If the problematic pool is optional (with the ;optional option specified), the pool is discarded.

To demonstrate how you can use these options in the `pool.properties` file, consider the following example:

```
$MASTERAGENTS;optional
POOL1
POOL2;skip
POOL3;optional;skip
POOL4;optional
```

Case 1: POOL1 and POOL4 exist, MASTERAGENTS does not exist

- POOL2;skip is not considered at all.
- POOL3;optional;skip is not considered at all because the ;skip option overrides the ;optional option.
- MASTERAGENTS;optional is the problematic pool and is optional and therefore not considered by the agent.
- POOL1 is not a problematic pool.
- POOL4 is not a problematic pool.

Outcome: The agent goes online and is inserted in POOL1 and POOL4.

Case 2: POOL1 does not exist, MASTERAGENTS and POOL4 exist

- POOL2;skip is not considered at all.
- POOL3;optional;skip is not considered at all because the ;skip option overrides the ;optional option.
- MASTERAGENTS;optional is not a problematic pool.
- POOL1 is the problematic pool and is mandatory and cannot be discarded.
- POOL4 is not a problematic pool.

Outcome: The agent goes offline and is not inserted in any of the pools.

Revoking and reissuing a JSON Web Token

Steps to revoke and reissue a JWT

To revoke a JSON Web Token (JWT), delete the workstation definition to which the JWT is associated from the database. You can perform this operation from the Dynamic Workload Console or from the command line. To delete the agent from the command line, perform the following steps:

1. Open a shell session.
2. Launch the composer script.
3. Type the following command:

   ```
   delete workstation workstation_name
   ```

   where

   workstation_name

   is the name of the agent whose JWT you want to revoke.

For more information about the delete command, see delete *(on page)*.

From the Dynamic Workload Console, you can perform the same operation as follows:

1. Log in to the Dynamic Workload Console.
2. Click **Design** > **Workload Designer**.
3. Select an engine.
4. Click the **Workstation** item card to display all existing workstations.
5. Select the workstation to be deleted.
6. Click **Delete**.

If you want the agent to authenticate with JWT again, download a new JWT to the agent using the AgentCertificateDownloader script.

Consider the following example:

```
./AgentCertificateDownloader.sh --wauser MDMAdmin --wapassword 125784gtrOLK8542Mnfdw!
--jwt true -tdwbhostname Saturn -tdwbport 37116
```

For more information about the AgentCertificateDownloader script, see Certificates download to dynamic agents and fault-tolerant agents - AgentCertificateDownloader script *(on page 338)*.

Chapter 8. Configuring a remote command-line client

To configure a remote command-line client, perform the following steps:

1. Log on as Administrator on Windows operating systems, or as root on UNIX and Linux operating systems, on the machine where the remote command-line client is installed with a fault-tolerant agent.
2. Open the `localopts` configuration file in the fault-tolerant agent instance.
3. Complete the `# Attributes for CLI connections` configuration section to connect the remote command-line client to the command-line server in the master domain manager:

   HOST

   The IP address or host name of the workstation where the master domain manager is installed.

   PROTOCOL

   The protocol that is used by the command-line client to connect to the workstation where the master domain manager is installed. The possible values are *http* and *https*. The default protocol that is used by the command-line client to establish a connection with the master is *https*.

   PORT

   The HTTP or HTTPS port number that is used to connect to the workstation where the master domain manager is installed. This port number must match the values that are defined for the master domain manager instance.

   TIMEOUT

   The timeout in seconds to wait for a master domain manager response.

   CLISSLSERVERAUTH

   Specify whether or not the connection to the master domain manager is SSL or not. If you set this value to `true`, perform the steps described in Configuring SSL connection between remote command-line client and master domain manager *(on page 149)*.

   CLISSLSERVERCERTIFICATE

   Specify only if *CLISSLSERVERAUTH* is set to `true`. The absolute path of the .arm file of the server public certificate. For more information about this value, see Configuring SSL connection between remote command-line client and master domain manager *(on page 149)*.

   CLISSLTRUSTEDDIR

   Specify only if *CLISSLSERVERAUTH* is set to `true`. The path of all the .arm files that the remote CLI must trust. For more information about this value, see Configuring SSL connection between remote command-line client and master domain manager *(on page 149)*.

   DEFAULTWS

   The master domain manager workstation name.

   USEROPTS

   The file that contains the user name and password to use to connect to the master domain manager workstation. This user must be a valid user that is listed in the `Security` file on the master domain manager.
4. Save the `localopts`.
5. Restart the fault-tolerant agent processes to accept the `localopts` changes.

Configuring SSL connection between remote command-line client and master domain manager

Before starting with the procedure to configure the SSL connection between the remote command-line client and the master domain manager, ensure that you set the *CLISSLSERVERAUTH* property to `true` in the *localopts* file of the fault-tolerant agent instance.

To configure a remote command-line client to connect to a master domain manager in SSL mode, perform the following steps:

1. Extract the certificate on the master domain manager instance by running the following procedure:
    a. Log on as Administrator on Windows operating systems, or as root on UNIX and Linux operating systems, on the machine where the master domain manager is installed.
    b. Extract the `server.crt` base 64 certificate by running:

```
keytool -export
-alias server
-rfc
-file server.crt
-keystore path>/TWSServerKeyFile.p12
-storepass default
```

    where <*path*> is one of the following:

    On Windows systems

    <TWA_home>`\usr\servers\engineServer\resources\security\TWSServerKeyFile.p12`

    On UNIX systems

    <*TWA_DATA_DIR*>`/usr/servers/engineServer/resources/security/TWSServerKeyFile.p12`

2. Log on as Administrator on Windows operating systems, or as root on UNIX and Linux operating systems, on the machine where the remote command-line client is installed with a fault-tolerant agent.
3. Perform a binary FTP of the `server.crt` certificate from the machine where you installed the master domain manager instance to the machine where you installed the remote command-line client in the directory `<FTA_INST_DIR>\ssl`.
4. Rename the `<FTA_INST_DIR>\ssl\server.crt` file to `<FTA_INST_DIR>\ssl\server.arm`.
5. Open the `localopts` configuration file in the fault-tolerant agent instance.
6. Complete one of the following attributes in the `# Attributes for CLI connections` configuration section and perform the actions:

    CLISSLSERVERCERTIFICATE

    Specify the absolute path of the `server.arm` file on the fault-tolerant agent machine. In this example, `<FTA_INST_DIR>\ssl\server.arm`.

    CLISSLTRUSTEDDIR

    Specify the path of the directory that contains all the `certificates.arm` files also the `<FTA_INST_DIR>\ssl\server.arm` that the remote command-line client can trust.

    Note: Do not set simultaneously the *CLISSLSERVERAUTH* and *CLISSLTRUSTEDDIR* values. For more information about the SSL configuration, see Connection security overview *(on page)*.

7. Save the `localopts` file.
8. Restart the fault-tolerant agent processes to accept the `localopts` changes.

Chapter 9. Configuring a z-centric agent on Windows operating systems

After you install a z-centric agent on a Windows operating system with a local or domain account, perform the following configuration steps:

1. Stop the dynamic agent.
2. From the **Start** menu, click **Administrative Tools > Services**.
3. Edit the properties of the following service by double-clicking on its name: `IBM Common Platform Agent: tws_cpa_agent_`*`TWS_user`*, where *`TWS_user`* is the name of the user for which IBM Workload Scheduler was installed (the name you supplied during installation).
4. Click label **Log On**.
5. Click **Log on as: Local System account**.
6. If you plan to run interactive jobs, check mark **Allow service to interact with desktop**.
7. Click **OK**.
8. From the **Start** menu, click **Administrative Tools > Local Security Policy**.
9. Remove the following permissions from the user created when you installed the z-centric agent:
    - Act as part of the operating system.
    - Log on locally.
    - Log on as batch.
10. Restart the dynamic agent.

Chapter 10. Adding a feature

Use the **twsinst** script to add the following feature to the IBM Workload Scheduler agent in your distributed or end-to-end network:

Add the Java™ run time to an agent

During the installation or the upgrade of the agent you might have chosen not to add the Java™ run time that supports the running of job types advanced options. This option provides your agent with the following capabilities:

- Run job types with advanced options, both those types supplied with the product and the additional types implemented through the custom plug-ins.
- Enable the capability to run remotely, from the agent, the dynamic workload broker resource command on the server.

If you later decide that you require this function, you can add the Java™ run time separately, as described below.

Procedure

To modify agents by using the twsinst script, perform the following steps:

On Windows™ operating systems

1. Download the eImage for your operating system. See Downloading installation images on your workstation *(on page 157)*.
2. Log in as administrator on the workstation where you want to upgrade the product.
3. From the *root*/TWS/*operating_system* directory of the eImage, run **twsinst** by using the synopsis described below.

Note: twsinst for Windows™ is a Visual Basic Script (VBS) that you can run in CScript and WScript mode, for example:

```
cscript twsinst -modify -uname username
-password user_password -acceptlicense yes
-addjruntime true
```

On UNIX™ and Linux™ operating systems

1. Download the eImage according to the operating system. See Downloading installation images on your workstation *(on page 157)*.
2. From the *root*/TWS/*operating_system* directory, run the **twsinst** script by using the synopsis described below.

A successful modify by using the **twsinst** script issues the return code RC = 0. If the operation fails, to understand the cause of the error, see Analyzing return codes for agent installation, upgrade, restore, and uninstallation *(on page 280)*.

Synopsis:

On Windows™ operating systems:

```
 -addjruntime true
 [-inst_dir install_directory]
 [-recovInstReg boolean]
```

On UNIX™ and Linux™ operating systems

Show command usage and version

```
./twsinst -u | -v
```

Modify an instance

```
./twsinst -modify -uname user_name
 -acceptlicense yes|no
 -addjruntime true
 [-inst_dir install_directory]
 [-recovInstReg boolean]
```

-acceptlicense *yes*|*no*

Specify whether or not to accept the License Agreement.

-addjruntime *true*

Adds the Java™ run time to run job types with advanced options to the agent. The run time environment is used to run application job plug-ins on the agent and to enable the capability to run remotely, from the agent, the dynamic workload broker resource command on the server. With the `-modify` option, the only valid value for this parameter is **true**.

This option is applicable to both fault-tolerant agents and dynamic agents.

-inst_dir *install_directory*

The installation directory for IBM Workload Scheduler. The default is the home directory of the user for which IBM Workload Scheduler is being installed.

-modify

Modifies an existing agent that was installed by using **twsinst**.

-password *user_password*

Windows™ operating systems only. The password of the user for which you are upgrading IBM Workload Scheduler.

-recovInstReg *boolean*

Select this option to recover workstations that have corrupt registry files without reinstalling the product. If you specify this option, IBM Workload Scheduler re-creates the installation registries. Valid values are **true** and **false**. The default value is **false**.

You can use this option also to recover registry files in a cluster environment; in this case you can run the command on any node of the cluster and not necessarily on the node where you installed IBM Workload Scheduler. This is useful when the cluster node where the product is installed is unavailable or in an inconsistent state.

-uname *username*

The name of the user for which IBM Workload Scheduler is being updated. The software is updated in this user's home directory. This user name is not to be confused with the user that performs the upgrade.

Part IV. Upgrading

How to upgrade IBM Workload Scheduler to the current version.

Overview

When upgrading your IBM® Workload Scheduler environment, it is a good practice to start with the upgrade of the Dynamic Workload Console first. If you upgrade the console to the new product version level, you can then use it to verify that your environment is working after upgrading the remaining components.

The upgrade procedure varies depending on the product version you currently have installed:

- if you have installed version 9.5.0.*x* or 10.*x*.*x* and want to upgrade to the General Availability version with a **direct upgrade** procedure, see Performing a direct upgrade from v 9.5.0.x or v 10.x.x to v 10.2.3 *(on page 162)*.
- if you have installed version 9.5.0.*x* or 10.*x*.*x* and want to upgrade to the General Availability version with a **parallel upgrade** procedure, see Parallel upgrade from version 9.5.0.x or 10.x.x to version 10.2.3 *(on page 177)*. This procedure might be useful when you have you have some of your components installed on operating systems which are no longer supported in version 10.2.3 and therefore cannot perform a direct upgrade.
- if you have installed version 9.4.0.*x* and want to upgrade to version 10.2.3. In this case, only a **parallel upgrade** is supported. For more information, see Parallel upgrade from version 9.4.0.x to version 10.2.3 *(on page 221)*.

In a **direct upgrade procedure from version 9.5.0.*x* or 10.*x*.*x***, you upgrade the Dynamic Workload Console and its database, then upgrade the dynamic domain manager and its backups and the database, then master domain manager and its backups and the database, and finally the domain managers and their backups, and the agents.

In a **parallel upgrade procedure from version 9.5.0.*x* or 10.*x*.*x***, you upgrade WebSphere Application Server Liberty, upgrade the Dynamic Workload Console and its database, then upgrade the database for the server components and install a new dynamic domain manager and master domain manager configured as a backup, then switch them to become the master. You then upgrade agents and domain managers.

In a **parallel upgrade procedure from version 9.4.0.*x***, you install the Dynamic Workload Console at v 10.2.3. You then upgrade the database tables for the server components and their backups and install a new backup dynamic domain manager, switch the manager to the new backup, install a new backup and switch back the manager capabilities, so that the newly installed backup dynamic domain manager becomes the current dynamic domain manager.

You then proceed to running the serverinst script to install a version 10.2.3 master domain manager configured as a backup. The installation process is able to detect the presence of an existing master domain manager and automatically configures the second one as the backup master domain manager. The new backup master domain manager is configured to point to the existing database instance. You then perform a switch with the previous version master domain manager, so that the newly installed backup master domain manager becomes the current active master domain manager.

You then install a second master domain manager to act as the new backup master domain manager. Each Dynamic Workload Console, backup dynamic domain manager, dynamic domain manager, master domain manager and backup master domain manager installation requires its own installation of WebSphere Application Server Liberty Base. The upgrade process concludes with upgrading agents. Agents can be upgraded with minimal disruption to scheduling activities.

During the master domain manager upgrade process, the license model to be applied to the environment is defined. The license model determines the criteria by which your license compliance is calculated. The following pricing models are supported: **byWorkstation**, **perServer**, **perJob**. The default value is **perServer**. To determine the current value of this global option, enter the following command: **optman show ln** or **optman show licenseType**. To modify the pricing model, use the **optman chg ln** or **optman chg licenseType** command. For more information about licensing, see License Management in IBM License Metric Tool *(on page)*.

Using the new features introduced with the latest release creates new records in the database which are not compatible with previous versions and therefore you cannot roll back your environment to a previous version.

If you upgrade IBM® Workload Scheduler to version 10.2.*x*, and the IBM® Workload Scheduler database was created with DB2, change the DB2 configuration parameter EXTENDED_ROW_SZ to ENABLE, or create a new buffer pool and table space

with a page size of 16 kilobytes and migrate the tables to the new table space. For more information, see Error in upgrading the IBM Workload Scheduler database when using a DB2 database *(on page 284)*.

Before upgrading, ensure that you have stopped workload processing on the master domain manager.

If you have previously customized the `tws_env` script, merge your changes into the new version of the script. Ensure you do not overwrite the parameters related to OpenSSL libraries during the merge.

Choosing how to upgrade your network

After upgrading the Dynamic Workload Console, there are different approaches to upgrading the remaining components in your IBM Workload Scheduler environment. Because IBM Workload Scheduler supports compatibility with earlier versions, after upgrading the console, you can decide to proceed with upgrading in one of the following ways, depending on the type of your network:

Top-down

Upgrade components in the following order:

1. backup domain managers and domain managers
2. dynamic domain managers
3. backup master domain manager
4. master domain manager
5. agents

This order ensures that events involving folders are correctly managed by the master domain manager and sent to agents at a supported version level.

When you have a backup master domain manager at the V9.5 Fix Pack 2, or later, but the master domain manager is still at a previous product version level, problems can occur when monitoring objects that support the definition in a folder such as, prompts, workstations, and resources, as well as objects that contain the workstation in their object identifier, for example, job streams. More specifically these objects are not displayed in the results of the monitoring query on the plan if you use filters in your query. To solve this problem, upgrade the master domain manager to the V9.5 Fix Pack 2 level, or later, and then run planman resynch.

Many of the new functions that are introduced in the current version become available for each agent as it is upgraded. The disadvantage is that the same functions are not available to all agents at the same time.

Bottom-up

Upgrade components in the following order:

1. agents
2. backup domain managers and domain managers
3. dynamic domain managers
4. backup master domain manager
5. master domain manager

The new functions that are introduced in the current version are not available until the whole network is upgraded.

In the typical upgrade procedures documented in this manual, the top-down order is used.

Note: Due to new support of the UPN Windows user, if you have Windows domain users that are defined in the logon fields as `domain\username`, after performing an upgrade to this version, update the `Security` file before starting the IBM Workload Scheduler instance. Insert the escape character '\' before the '\' character in the `domain\username` value. For example, if you use the `MYDOMAIN\user1` value in the logon field, after the upgrade, in the `Security` file you must update the line in following way:

```
.............
logon=MYDOMAIN\\user1
.............
```

For details, see Configuring the Security File *(on page)*.

Migrating custom events

When you perform an upgrade, custom events are not migrated. Therefore you must add custom events by following the manual procedure described below:

1. On the master domain manager, create a new XML file `<file_name>` where you can save custom events:

   ```
   $ evtdef dumpdef <file_name>
   ```

2. Run the `switchmgr` command to switch from the master domain manager to the backup master domain manager.
3. Copy the XML file created in step 1 on the backup master domain manager.
4. Load the custom event definition on the backup master domain manager by running the following command:

   ```
   $ evtdef loaddef <file_name>
   ```

   Where `<file_name>` is the name of the XML file that you copied from the master domain manager and saved on the backup master domain manager.
5. Stop WebSphere Application Server Liberty on the Dynamic Workload Console.
6. Start WebSphere Application Server Liberty on the Dynamic Workload Console by running the following command:

   ```
   <DWC_HOME>/appservertools/startAppServer.sh -directclean
   ```

Chapter 1. Downloading installation images on your workstation

Steps to take when downloading images on your workstation.

Complete the following procedure to download the installation images to upgrade your environment to the latest level:

1. Ensure that your workstation has sufficient space to store both the files you download from IBM Fix Central and the extracted installation image. For more information about system requirements, see IBM Workload Scheduler Detailed System Requirements and Dynamic Workload Console Detailed System Requirements. To install the product, download all the required images from IBM Fix Central. The zip contains both the General Availability 10.2.3 image and the latest fix pack image, if available.
2. Download the installation images from IBM Fix Central.
3. Extract the installation image from the downloaded file and verify that the installation image is complete.

Chapter 2. Upgrading from the CLI

Upgrade IBM Workload Scheduler from the command-line interface.

The upgrade procedure varies depending on the product version you currently have installed:

- if you have installed version 9.5.0.*x* or 10.*x.x* and want to upgrade to the General Availability version with a **direct upgrade** procedure, see Performing a direct upgrade from v 9.5.0.x or v 10.x.x to v 10.2.3 *(on page 162)*.
- if you have installed version 9.5.0.*x* or 10.*x.x* and want to upgrade to the General Availability version with a **parallel upgrade** procedure, see Parallel upgrade from version 9.5.0.x or 10.x.x to version 10.2.3 *(on page 177)*. This procedure might be useful when you have you have some of your components installed on operating systems which are no longer supported in version 10.2.3 and therefore cannot perform a direct upgrade.
- if you have installed version 9.4.0.*x* and want to upgrade to version 10.2.3. In this case, only a **parallel upgrade** is supported. For more information, see Parallel upgrade from version 9.4.0.x to version 10.2.3 *(on page 221)*.

Certificates are now required when installing or upgrading IBM® Workload Scheduler. You can no longer install nor upgrade IBM® Workload Scheduler without securing your environment with certificates. The required certificates are:

- ca.crt
- tls.key
- tls.crt

For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).

Before upgrading

Before starting to upgrade the product, verify that your network has the minimum required supported versions of the operating system, product, and database.

Supported operating systems

To produce a dynamic report that lists the supported operating systems, click Supported operating systems.

For a complete list of system requirements (disk spaces, temporary spaces and RAM usage), see IBM Workload Scheduler Detailed System Requirements.

Supported databases

For an up-to-date list of supported databases, run the Detailed Software Requirements report and select the Prerequisites tab.

Product level prerequisites for master domain manager and its backup, dynamic domain manager and its backup, and agents

Before you start the upgrade, verify that your environment has the required product level prerequisites. For a complete list of product level prerequisites, see IBM Workload Scheduler Detailed System Requirements.

User authorization requirements

Before starting to upgrade, verify that the user running the process has the following authorization requirements:

Windows™ operating systems

If you set the Windows User Account Control (UAC), your login account must be a member of the Windows™ **Administrators** group or domain administrators with the right, **Act as Part of the Operating System**.

If you set the Windows User Account Control (UAC) on the workstation, you must run the installation as **administrator**.

UNIX™ and Linux™ operating systems

If the component was installed with root privileges, **root** access is required. If you performed a **no-root installation**, specify the same user used for installing the component.

SSL mode configuration

If the IBM® Workload Scheduler environment is configured in SSL mode, ensure one of the following conditions is met in the `localopts` file before you upgrade master domain manager, backup master domain manager, dynamic domain manager, or fault-tolerant agents to Version 10.2.3 or later:

- the **ssl version** parameter is set to `atleast.TLSv1.2` OR
- the **ssl cipher** parameters is set to a high value.

For more information about the `localopts` file, see Setting local options *(on page)*.

Securing your environment with certificates

Starting from version 10.2.1, using certificates is mandatory when installing or upgrading the product. You can use default certificates, generated automatically by the product with the password you specify, or you can define your own custom certificates. For more information, see Enhanced security for default certificates *(on page)*.

In the typical upgrading procedures described in this section, default certificates are used.

Upgrading to 10.1 Fix Pack 1 or later using custom certificates

In 10.1 FP1 version, the JWT feature has been introduced. Performing an upgrade of the master domain manager to 10.1 FP1 from any previous version, can potentially cause problems with JWT functionality if the master domain manager is using custom certificates with a custom label.

master domain manager. Therefore, it is required to also add the public information only of the custom certificate of the master domain manager (the file that was added in the `TWSServerTrustFile.p12` file on the master domain manager) in the **TWSClientKeyStore** file of the agent.

Support for OpenSSL 3.0.*x* libraries from UNIX operating systems

If you install the master domain manager on recent UNIX operating systems, you can use the OpenSSL 3.0.*x* libraries provided with the operating system. The list of UNIX operating systems whose libraries you can use is as follows:

- Ubuntu 22
- AIX 7.3
- Red Hat 9

To ensure IBM® Workload Scheduler uses these libraries, always launch the installation or upgrade procedure from a brand new shell. You can also check the OpenSSL library currently in use with the which openssl command and check the OpenSSL version with the openssl version command.

Downloading installation images

Before starting to upgrade, download the installation images. For further information, see Downloading installation images on your workstation *(on page 157)*

Scanning system prerequisites for IBM Workload Scheduler

Before installing or upgrading the product, IBM Workload Scheduler automatically runs a scan on your system.

When installing IBM Workload Scheduler using the serverinst script, the script first runs the scanner to verify system prerequisites. For more information about prerequisites, see Download Documents, System Requirements, Release Notes *(on page)*.

Note: To ensure that the prerequisite scan process does not fail, verify that the `bc` executable is present on the local system and that it is set in the PATH environment variable. If you do not want to install the `bc` executable, you can skip the prerequisites check by using the **skipcheckprereq** parameter when running the serverinst and twsinst parameters. For more information about the `bc` executable, see bc, an arbitrary precision calculator language. For more information about installation commands, see Server components installation - serverinst script *(on page 310)* and Agent installation parameters - twsinst script *(on page 84)*.

Having an environment that meets the product system requirements ensures that an installation or upgrade succeeds without any delays or complications.

The scan verifies that:

- The operating system is supported for the product.
- On UNIX™ operating systems, the necessary product libraries are installed.
- There is enough permanent and temporary disk space to install both the product and its prerequisites.
- There is enough memory and virtual memory.

Note: The scan verifies only that the environment meets the requirements of IBM Workload Scheduler. It does not check the requirements for other components, such as DB2®.

If any of these checks fails, IBM Workload Scheduler returns an error message.

The log files for the server components are located in:

On Windows™ operating systems:

<*TWA_home*>`\logs\serverinst<`*`version_number`*`>.log`

On UNIX™ and Linux™ operating systems:

<*TWA_DATA_DIR*>/installation/logs/serverinst<*version_number*>.log

The log files for the Dynamic Workload Console are located in:

On Windows™ operating systems:

<*DWC_home*>\logs\dwcinst<*version_number*>.log

On UNIX™ and Linux™ operating systems:

<*DWC_DATA_dir*>/installation/logs/dwcinst<*version_number*>.log

The log files for the agents are located in:

On Windows™ operating systems:

<*TWA_home*>\logs\twsinst<*interp*><*user_name*><*version_number*>.log

On UNIX™ and Linux™ operating systems:

<*TWA_DATA_DIR*>/installation/logs/
twsinst<*interp*><*user_name*><*version_number*>.log

You can decide to rerun the installation or upgrade without executing the prerequisite scan. If you set the **-skipcheckprereq** parameter to `true` when performing the installation, the installation script does not execute the prerequisite scan. If a problem occurs, an error is displayed, the component is installed or upgraded, but might not work. For more information about the `-skipcheckprereq` parameter in all installation scripts, see Reference *(on page 300)*.

Connecting the Dynamic Workload Console to a new node or database

Move Dynamic Workload Console data to a new node or database by exporting data to an XML file to be imported in the new instance.

If you want to move the Dynamic Workload Console to a new node or database, you need to export the settings from an existing instance and create an XML file that can be imported into another Dynamic Workload Console node or database. If in your current environment you are using Derby, you can use this procedure to move to another supported database, because Derby is no longer supported starting from version 10.2.3.

Note: The migration of the roles from the Dynamic Workload Console Version 9.4 to Version 10.2.3 is not supported. You have to recreate the roles in the latest version.

To export the Dynamic Workload Console settings from the previous installation, perform the following procedure:

1. From the navigation toolbar, click **Administration** > **Manage Settings**.
2. In the Manage Settings panel, click **Export Settings** and save the XML file to a directory of your choice.
3. Optionally, edit the file using an XML editor and save it.
4. Optionally, export your custom boards: from the dashboard to be exported, click on the options menu next to the name of the dashboard and select **Export**. A JSON file is downloaded.
5. Browse to the `datasource_<db_vendor>.xml` file located in one of the following paths:

   On UNIX operating systems

   DWC_home/usr/servers/dwcServer/configDropins/templates

   On Windows operating systems

   DWC_home\usr\servers\dwcServer\configDropins\templates

6. Copy the `datasource_<db_vendor>.xml` to the path for your operating system:

   On UNIX operating systems

   DWC_DATA_dir/usr/servers/dwcServer/configDropins/overrides

   On Windows operating systems

   DWC_home\usr\servers\dwcServer\configDropins\overrides

7. Configure the `datasource_<db_vendor>.xml` file based on the specifics of your environment.

8. Only if you are moving from an Oracle database to a different database, browse to the following files:
   - *DWC_DATA_dir*`>/usr/servers/dwcServer/apps/DWC.ear/DWCRest.war/META-INF/orm.xml`
   - *DWC_DATA_dir*`/usr/servers/dwcServer/apps/DWC.ear/Reporting.war/META-INF/orm.xml`
   - *DWC_DATA_dir*`/usr/servers/dwcServer/apps/DWC.ear/TWSWebUI.war/META-INF/orm.xml`

   and replace the contents of the **<schema>** tag with **<schema>TDWC</schema>**.
9. Copy the settings file generated from the procedure to the workstation where the new Dynamic Workload Console is to be installed.

You can now proceed with the upgrade, either direct or parallel, based on the procedure you have chosen. You will import the settings and boards when you install or upgrade the new Dynamic Workload Console.

If you need to connect the master domain manager to a new database, see Connecting the master domain manager to a new database *(on page)*.

Performing a direct upgrade from v 9.5.0.*x* or v 10.*x*.*x* to v 10.2.3

Detailed steps to perform a direct upgrade from version 9.5.0.*x* or v 10.*x*.*x* to version 10.2.3

To upgrade your environment using a direct upgrade procedure, perform the following steps:

1. Converting default certificates *(on page 163)*, if you are using default certificates in your current environment. Use this procedure to convert the certificates from the JKS to the PEM format, then copy them to the workstations where you plan to install the server components (dynamic domain manager and its backups, master domain manager and its backups) and the Dynamic Workload Console.

   Certificates are now required when installing or upgrading IBM® Workload Scheduler. You can no longer install nor upgrade IBM® Workload Scheduler without securing your environment with certificates. The required certificates are:
   - ca.crt
   - tls.key
   - tls.crt

   For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).
2. Upgrading WebSphere Application Server Liberty *(on page 165)* on the workstations hosting Dynamic Workload Console and the server components (dynamic domain manager and its backups, master domain manager and its backups).
3. Performing a direct upgrade of the Dynamic Workload Console and its database *(on page 166)*
4. Performing a direct upgrade of the dynamic domain manager, its backups, and their database *(on page 167)*
5. Performing a direct upgrade of the backup master domain manager and its database *(on page 169)*
   a. Switching the master domain manager to the upgraded backup master *(on page 171)*
   b. Making the switch permanent *(on page 171)*
6. Performing a direct upgrade of the master domain manager *(on page 172)*
   a. Switching back to the master domain manager from the backup master domain manager *(on page 174)*
   b. Making the switch permanent *(on page 175)*
7. Upgrading agents and domain managers *(on page 175)*

Environment with custom certificates

If you have version 9.5 installed with custom certificates, then after upgrading to 10.2 you must ensure that the parameters and the name of the relevant certificates in the **localopts** file are correct.

If you have previously used certificates generated with OpenSSL, check the paths in the following section:

- For Open SSL, check:
  - SSL key
  - SSL certified
  - SSL key pwd
  - SSL CA certified
  - SSL random seed

If you have used GSKit, the relevant parameters are automatically migrated to the new OpenSSL parameters:

- **SSL Version**
- **SSL Ciphers**
- **CLI SSL Ciphers**
- **CLI SSL Version**

If the value of these fields corresponds to an incorrect path, then stop the WebSphere Application Server Liberty Base, make the necessary changes, and refer to the client's certificates, and then Restart.

For more information, see Setting local options *(on page)*.

Converting default certificates

Procedure to extract and convert default certificates generated in your current version prior to upgrading.

If you are using default certificates, extract and convert them before you start the upgrade. Perform the following steps:

1. Set the IBM® Workload Scheduler environment, as described in Setting the environment variables *(on page 138)*.
2. To ensure the keytool and openssl commands start correctly on all operating systems, browse to the folder where the keytool and openssl commands are located and launch the commands as follows:

   ```
   cd <TWS_DIR>/JavaExt/jre/jre/bin
   ```

   ```
   ./keytool -importkeystore -srckeystore TWSServerKeyFile.jks -destkeystore
   <path_of_extracted_certs>/server.p12 -deststoretype pkcs12
   ```

   ```
   cd <TWS_DIR>/tmpOpenSSL64/1.1/bin/openssl
   ```

   ```
   ./openssl pkcs12 -in <path_of_extracted_certs>/server.p12 -out
   <path_of_extracted_certs>/tls.tot
   ```

   The location of the `TWSServerKeyFile.jks` varies depending on the IBM® Workload Scheduler version you have currently installed, as follows:

   versions 9.5 and later

   TWA_DATA_DIR`/usr/servers/engineServer/resources/security`

   versions 9.4 and earlier

   TWA_home`/WAS/TWSProfile/etc`

3. Open the `tls.tot` file with any text editor.
4. From the `tls.tot` file, copy the private key to a new file named `tls.key`.
   The `tls.key` file must be structured as follows:

```
----BEGIN ENCRYPTED PRIVATE KEY----
<private_key>
----END ENCRYPTED PRIVATE KEY----
```

Note: Insert a carriage return after each key, so that an empty line is inserted after each key.

5. From the `tls.tot` file, copy the public key to a new file named `tls.crt`.
   The `tls.crt` file must be structured as follows:

```
----BEGIN CERTIFICATE----
<public_key>
----END CERTIFICATE----
```

Note: Insert a carriage return after each key, so that an empty line is inserted after each key.

6. Copy the contents of the `tls.crt` file into a new file named `ca.crt`. If you want to upgrade a dynamic domain manager, also copy the contents of the `tls.crt` file into another new file named `jwt.crt`.
7. Create a file named `tls.sth` containing the passphrase you have specified for creating the .`p12` certificate in step 2 *(on page 163)*, encoded in `base64` format. To create the `tls.sth` file, use the following command:

```
./secure -password your_password -base64 e -out
<path_of_extracted_certs>/tls.sth
```

   If you are using a version earlier than 10.*x*, you can find the secure script in the installation package of the 10.2.3 version you are upgrading to. You can launch the script from on of the following paths:

   master domain manager and agent

```
<10.2.3_extracted_image_dir>/TWS/<interp>/Tivoli_LWA_<interp>/TWS/bin
```

   Dynamic Workload Console

```
<10.2.3_extracted_image_dir>/DWC/<interp>/bin
```

   where

   <interp>

   is the operating system you are installing on

   As an alternative, you can use the following command on UNIX workstations:

```
echo -n "passwordToEncode" | base64 >> tls.sth
```

8. Browse to the GSKit folder and extract the client certificates from the *TWA_DATA_DIR*`/ssl/GSKit` folder by running the following commands, depending on the IBM® Workload Scheduler version you have currently installed:

```
cd <TWS_DIR>/tmpGSKit64/8/bin
```

   versions 9.5 and later

```
./gsk8capicmd_64 -cert -extract -db <TWA_DATA_DIR>/ssl/GSKit/TWSClientKeyStore.kdb
-stashed -label client -target <path_of_extracted_certs>/client.crt
```

   versions 9.4 and earlier

```
./gsk8capicmd_64 -cert -extract -db <TWS_DIR>/ssl/GSKit/TWSClientKeyStore.kdb
-stashed -label client -target <path_of_extracted_certs>/client.crt
```

9. Create a folder named `additionalCAs` in the folder where you extracted the certificates and move the `client.crt` file created in step 8 *(on page 164)* to the `additionalCAs` folder.
10. Insert the `client.crt` in the `additionalCAs` folder when providing the certificates to the installation script with the **sslkeysfolder** parameter.

11. Assign the correct permissions (755) and ownerships to extracted certificates, as follows:

```
chmod -R 755 <path_of_extracted_certs>
```

You have now extracted and converted your certificates for use with version 10.2.3.

You can now optionally upgrade WebSphere Application Server Liberty, as described in Upgrading WebSphere Application Server Liberty *(on page 165)*. When upgrading IBM® Workload Scheduler components in upcoming steps, provide the path to the folder where you extracted the certificates using the **sslkeysfolder** parameter when running the installation scripts. For more information about the installation scripts, see Reference *(on page 300)*.

Upgrading WebSphere Application Server Liberty

This is an optional step you might want to perform before you upgrade the Dynamic Workload Console and the master components.

Upgrading WebSphere Application Server Liberty Base

Perform the following steps to upgrade WebSphere Application Server Liberty Base to the latest supported version on the workstations hosting the Dynamic Workload Console and the server components (dynamic domain manager and its backups, master domain manager and its backups).

1. Download WebSphere Application Server Liberty Base from Recommended updates for WebSphere Application Server Liberty.
   Each WebSphere Application Server Liberty Base image is packaged as a `.jar` file named `wlp-base-all-version.jar`.

   Check the release notes to ensure the latest WebSphere Application Server Liberty Base version is supported by IBM Workload Scheduler You can find the Release Notes at IBM Workload Scheduler Release Notes.
2. Stop the application server as described in Application server - starting and stopping *(on page)*.
   Also stop IBM® Workload Scheduler and all other applications running on the WebSphere Application Server Liberty Base instance.
3. Optionally create a backup of the current WebSphere Application Server Liberty Base instance in a directory different from the WebSphere Application Server Liberty Base installation directory.
4. Uninstall WebSphere Application Server Liberty Base.
5. Install WebSphere Application Server Liberty Base by extracting the archive file to a directory of your choice.

   On Windows operating systems

   ```
   java -jar liberty_download_dir\wlp-base-all-version.jar
   --acceptLicense install_dir
   ```

   On UNIX operating systems

   ```
   java -jar liberty_download_dir/wlp-base-all-version.jar
   --acceptLicense install_dir
   ```

   where:

   liberty_download_dir

   The directory where you downloaded WebSphere Application Server Liberty Base.

   install_dir

   The directory where you want to upgrade WebSphere Application Server Liberty Base.

   Note: Install the new WebSphere Application Server Liberty Base in the exact location of the previous WebSphere Application Server Liberty Base installation.

6. Restart the application server as described in Application server - starting and stopping *(on page)*.
   Also restart IBM® Workload Scheduler and all other applications running on the WebSphere Application Server Liberty Base instance.

You have now successfully upgraded WebSphere Application Server Liberty and can proceed to Performing a direct upgrade of the Dynamic Workload Console and its database *(on page 166)*, to Performing a direct upgrade of the backup master domain manager and its database *(on page 169)*, or to Performing a direct upgrade of the master domain manager *(on page 172)*.

Performing a direct upgrade of the Dynamic Workload Console and its database

Perform a direct upgrade of the Dynamic Workload Console from version 9.5.0.*x* to version 10.2.*x*. If you have several Dynamic Workload Console nodes in a cluster, upgrade all the nodes in the cluster.

When upgrading the IBM Workload Scheduler environment, it is a good practice to update the Dynamic Workload Console first. If you update the console, you can then use it to verify that your environment is working after updating the remaining components.

Note: If you are installing the Dynamic Workload Console version 10.2.3 or later, the Federator is also automatically installed. This component enables you to monitor your objects through the Orchestration Monitor page of the Dynamic Workload Console. For detailed information about how to configure and use the Federator, see Mirroring the z/OS current plan to enable the Orchestration Monitor *(on page)*.

If you are currently using Derby, you need to install a supported database and migrate your data. This is necessary because Derby is no longer supported as of version 10.2.3. For more information, see Connecting the Dynamic Workload Console to a new node or database *(on page 161)*.

1. Log in to the workstation where you plan to install the Dynamic Workload Console.
2. On UNIX™ operating systems, ensure that **umask** is set to **022**. To verify that **umask** is set to the correct value, from a command prompt, run the **umask** command. If the value is different from **022**, modify it by running the following command:

   ```
   umask 022
   ```

   On Windows operating systems, ensure you have the correct rights on the folder where you plan to install.
3. Download the installation images from IBM Fix Central.
4. Browse to the folder *image_location*.
5. If possible, stop all Dynamic Workload Console instances.
   If this is not possible, launch the configureDB script at a time when the Dynamic Workload Console is processing a low workload. If the configureDB script should fail because of conflicts with the Dynamic Workload Console, restart the script.
6. If your current version is earlier than 10.2.1, your certificates need to be updated before they can work with version 10.2.3. The update is performed automatically, but you need to provide the password for the certificates. Define an environment variable with name **JKS_SSL_PASSWORD** and set it to the password you defined for the certificates. You can optionally encrypt the password using the secure script. For more information about the secure script, see Optional password encryption - secure script *(on page 300)*.
7. To update the database version, run the following command:

   On Windows operating systems

   ```
   cscript configureDb.vbs --rdbmstype db_type --dbhostname db_hostname
   --dbport db_port --dbname db_name --dbuser db_user
   --dbpassword db_password --dbadminuser db_administrator
   --dbadminuserpw db_administrator_password
   ```

On UNIX operating systems

```
./configureDb.sh --rdbmstype db_type --dbhostname db_hostname
--dbport db_port --dbname db_name --dbuser db_user
--dbpassword db_password --dbadminuser db_administrator
--dbadminuserpw db_administrator_password
```

On z/OS operating systems

```
./configureDb.sh --rdbmstype db_type --dbhostname DB_hostname
--dbport db_port --dbname db_name --dbuser db_user
--dbadminuser DB_admin_user --dbadminuserpw DB_admin_pwd
--zlocationname zOS_location_containing_db --zbufferpoolname buffer_pool_in_zOS_location
```

For more information about all parameters and supported values of the `configureDb` command, see Database configuration - configureDb script *(on page 301)*.

8. Start the upgrade by launching the following command:

   On Windows operating systems

   ```
   cscript dwcinst.vbs --acceptlicense yes --inst_dir INST_DIR
   ```

   On UNIX operating systems

   ```
   ./dwcinst.sh --acceptlicense yes --inst_dir INST_DIR
   ```

   On z/OS operating systems

   ```
   ./dwcinst.sh --acceptlicense yes --inst_dir INST_DIR
   ```

   For further details about commands, see Reference *(on page 300)*.

9. If you had previously exported the Dynamic Workload Console, as described in Connecting the Dynamic Workload Console to a new node or database *(on page 161)*, you can now import them in the new Dynamic Workload Console from the **Administration** > **Manage Settings** menu. If you have a high availability configuration, import the settings on one node.
10. If you have copied any template `.xml` files from the `templates` folder to the `overrides` folder, check for any differences between the default `.xml` files just upgraded in the `templates` folder and the files you are using in the `overrides` folder. If any differences are present, update the files in the `overrides` folder accordingly. For example, in version 10.2.3, the following variables have been added in the `ssl_config.xml` file and must be added in the corresponding file, if present in the `overrides` folder:

    ```
    <jndiEntry id="keyStore.location" jndiName="keyStore.location" decode="false"
    value="${server.config.dir}resources/security/${keyStore.location}"/>
    <jndiEntry id="trustStore.location" jndiName="trustStore.location" decode="false"
    value="${server.config.dir}resources/security/${trustStore.location}"/>
    ```

You have now successfully upgraded the Dynamic Workload Console. You can now proceed to upgrade domain managers using the procedure described in Performing a direct upgrade of the dynamic domain manager, its backups, and their database *(on page 167)*.

Performing a direct upgrade of the dynamic domain manager, its backups, and their database

Complete this procedure to upgrade the dynamic domain manager and the backup dynamic domain manager from version 9.5.0.*x* to version 10.2.*x*.

Upgrade a dynamic domain manager and a backup dynamic domain manager from version 9.5.0.*x* to version 10.2.*x* by running the **serverinst** script. Launch the script on the workstation where the dynamic domain manager is running to upgrade the

dynamic domain manager, then launch the script on the workstation where the backup dynamic domain manager is running to upgrade the backup dynamic domain manager.

1. Log in to the workstation where you plan to install.
2. On UNIX™ operating systems, ensure that **umask** is set to **022**. To verify that **umask** is set to the correct value, from a command prompt, run the **umask** command. If the value is different from **022**, modify it by running the following command:

   ```
   umask 022
   ```

   On Windows operating systems, ensure you have the correct rights on the folder where you plan to install.
3. Download the installation images from IBM Fix Central.
4. Browse to the folder *image_location*/TWS/interp_name.
5. Stop all IBM® Workload Scheduler services and WebSphere Application Server Liberty, by running the following commands:

   ```
   conman stop; wait
   conman shut; wait
   conman ShutDownLwa
   stopappserver
   ```

6. To update the database version, run the following command:

   On Windows operating systems

   ```
   cscript configureDb.vbs –-componenttype=DDM --rdbmstype db_type --dbhostname db_hostname
   --dbport db_port --dbname db_name --dbuser db_user --dbpassword db_password --dbadminuser
   db_administrator --dbadminuserpw db_administrator_password
   ```

   On UNIX operating systems

   ```
   ./configureDb.sh –-componenttype=DDM --rdbmstype db_type --dbhostname db_hostname --
   dbport db_port --dbname db_name --dbuser db_user --dbpassword db_password --dbadminuser
   db_administrator --dbadminuserpw db_administrator_password
   ```

   For more information about the **configureDb** script, see Database configuration - configureDb script *(on page 301)*.
7. If your current version is earlier than 10.2.1, your certificates need to be updated before they can work with version 10.2.3. The update is performed automatically, but you need to provide the password for the certificates. Define an environment variable with name **JKS_SSL_PASSWORD** and set it to the password you defined for the certificates. You can optionally encrypt the password using the secure script. For more information about the secure script, see Optional password encryption - secure script *(on page 300)*.
8. Check your FIPS settings. Starting from version 10.2.1, FIPS is no longer supported. If FIPS is enabled in your current environment, specify the **enablefips** parameter when installing and set it to `false`. This will disable FIPS by changing the related keywords in the `localopts` and `ita.ini` files. For more information, see Q: My environment is FIPS compliant. What happens if I upgrade to version 10.2.3? *(on page 270)*. If FIPS is not enabled in your current environment, you can skip the **enablefips** parameter.
9. Start the installation launching the following command:

   On Windows operating systems

   ```
   cscript serverinst.vbs --acceptlicense yes --inst_dir INST_DIR
   ```

   On UNIX operating systems

   ```
   ./serverinst.sh --acceptlicense yes --inst_dir INST_DIR
   ```

   where *INST_DIR* is the directory where the component is installed. To find out the installation directory, see Finding out what has been installed in which IBM Workload Automation instances *(on page 25)*.

   Note: The **acceptlicense** and **inst_dir** parameters are required. All other parameters are ignored by the serverinst command, except for the following two optional parameters: **lang** and **skipcheckprereq**.

   For further details about the serverinst script, see Server components installation - serverinst script *(on page 310)*.
10. If you have copied any template `.xml` files from the `templates` folder to the `overrides` folder, check for any differences between the default `.xml` files just upgraded in the `templates` folder and the files you are using in the

`overrides` folder. If any differences are present, update the files in the `overrides` folder accordingly. For example, in version 10.2.3, the following variables have been added in the `ssl_config.xml` file and must be added in the corresponding file, if present in the `overrides` folder:

```
<jndiEntry id="keyStore.location" jndiName="keyStore.location" decode="false"
value="${server.config.dir}resources/security/${keyStore.location}"/>
<jndiEntry id="trustStore.location" jndiName="trustStore.location" decode="false"
value="${server.config.dir}resources/security/${trustStore.location}"/>
```

11. After the installation has completed, run the following commands to start up IBM® Workload Scheduler services and WebSphere Application Server Liberty:

```
conman start
conman startappserver
StartUpLwa
```

You have now successfully upgraded the dynamic domain manager and its backup. You can now proceed to Performing a direct upgrade of the backup master domain manager and its database *(on page 169)*.

Performing a direct upgrade of the backup master domain manager and its database

Performing a direct upgrade of the backup master domain manager.

Upgrade a backup master domain manager from v 9.5.0.*x* v 10.2.*x* by running the **serverinst** script.

1. Log in to the workstation where you plan to install.
2. On UNIX™ operating systems, ensure that **umask** is set to **022**. To verify that **umask** is set to the correct value, from a command prompt, run the **umask** command. If the value is different from **022**, modify it by running the following command:

```
umask 022
```

   On Windows operating systems, ensure you have the correct rights on the folder where you plan to install.
3. Download the installation images from IBM Fix Central.
4. Browse to the folder `<image_location>/TWS/interp_name`.
5. Stop all IBM® Workload Scheduler services and WebSphere Application Server Liberty, by running the following commands:

```
conman "stop;wait"
conman "stopappserver;wait"
conman "shut;wait"
ShutDownLwa
```

6. If your current version is earlier than 10.2.1, your certificates need to be updated before they can work with version 10.2.3. The update is performed automatically, but you need to provide the password for the certificates. Define an environment variable with name **JKS_SSL_PASSWORD** and set it to the password you defined for the certificates. You can optionally encrypt the password using the secure script. For more information about the secure script, see Optional password encryption - secure script *(on page 300)*.
7. To update the database version, run the following command:

   On Windows operating systems

   ```
   cscript configureDb.vbs --rdbmstype db_type --dbhostname db_hostname --dbport db_port --
   dbname db_name --dbuser db_user --dbpassword db_password --dbadminuser db_administrator --
   dbadminuserpw db_administrator_password
   ```

   On UNIX operating systems

```
./configureDb.sh --rdbmstype db_type --dbhostname db_hostname --dbport db_port --dbname
db_name --dbuser db_user --dbpassword db_password --dbadminuser db_administrator --
dbadminuserpw db_administrator_password
```

For more information about the configureDb script, see Database configuration - configureDb script *(on page 301)*.

8. Check your FIPS settings. Starting from version 10.2.1, FIPS is no longer supported. If FIPS is enabled in your current environment, specify the **enablefips** parameter when installing and set it to `false`. This will disable FIPS by changing the related keywords in the `localopts` and `ita.ini` files. For more information, see Q: My environment is FIPS compliant. What happens if I upgrade to version 10.2.3? *(on page 270)*. If FIPS is not enabled in your current environment, you can skip the **enablefips** parameter.
9. Start the installation launching the following command:

   On Windows operating systems

   ```
   cscript serverinst.vbs --acceptlicense yes --inst_dir INST_DIR
   ```

   On UNIX operating systems

   ```
   ./serverinst.sh --acceptlicense yes --inst_dir INST_DIR
   ```

   where *INST_DIR* is the directory where the component is installed. To find out the installation directory, see Finding out what has been installed in which IBM Workload Automation instances *(on page 25)*.

   For more information about the serverinst script, see Server components installation - serverinst script *(on page 310)*.

   Note: The **acceptlicense** and **inst_dir** parameters are required. You can also specify the following optional parameters:
   - **lang**
   - **work_dir**
   - **skipcheckprereq**
   - **enablefips**

   If you specify other parameters, they are ignored and the settings from the current instance are used instead.

10. After the installation has completed, run the following commands to start up IBM® Workload Scheduler services and WebSphere Application Server Liberty:

    ```
    conman start
    conman startappserver
    StartUpLwa
    ```

11. To link all fault-tolerant agents, type the following command:

    ```
    conman "link @!/@/@ "
    ```

12. If you have copied any template `.xml` files from the `templates` folder to the `overrides` folder, check for any differences between the default `.xml` files just upgraded in the `templates` folder and the files you are using in the `overrides` folder. If any differences are present, update the files in the `overrides` folder accordingly. For example, in version 10.2.3, the following variables have been added in the `ssl_config.xml` file and must be added in the corresponding file, if present in the `overrides` folder:

    ```
    <jndiEntry id="keyStore.location" jndiName="keyStore.location" decode="false"
    value="${server.config.dir}resources/security/${keyStore.location}"/>
    <jndiEntry id="trustStore.location" jndiName="trustStore.location" decode="false"
    value="${server.config.dir}resources/security/${trustStore.location}"/>
    ```

You have now successfully upgraded the backup master domain manager. You can now proceed to Switching the master domain manager to the upgraded backup master *(on page 171)*.

Switching the master domain manager to the upgraded backup master

To switch the back-level master domain manager to the upgraded backup master domain manager, complete the following procedure:

1. Switch to your upgraded backup master domain manager, which now becomes your current active master domain manager, by issuing the following command from either the Dynamic Workload Console or the **command line** of your back-level master domain manager:

   From the Dynamic Workload Console

   In the navigation tree, click **Monitoring and Reporting > Monitor Workload >** select the engine and the object type Workstation, click run and, in the table of results, select backup master domain manager workstation name, click **More Actions**, and select **Become Master Domain Manager**.

   From the command line of the back-level master domain manager

   Issue the following command:

   ```
   conman "switchmgr masterdm;new_mgr_cpu"
   ```

   where *new_mgr_cpu* is the backup master domain manager workstation name.

2. Switch the event processor from the back-level master domain manager to the backup master domain manager, by running the following command from either the Dynamic Workload Console or the **command line** of your back-level master domain manager:

   From the Dynamic Workload Console

   In the navigation tree, click **Monitoring and Reporting > Monitor Workload >** select the engine and the object type Workstation, click run and, in the table of results, select backup master domain manager workstation name, click **More Actions**, and select **Become Event Processor**.

   From the command line of the back-level master domain manager

   Issue the following command:

   ```
   conman "switcheventprocessor new_mgr_cpu"
   ```

   where *new_mgr_cpu* is the backup master domain manager workstation name.

Once you have switched the master domain manager to the upgraded backup master, you can make this switch permanent. For details, see Making the switch permanent *(on page 171)*.

For more detailed information about switching the master domain manager, see Short-term switch of a master domain manager *(on page)*

Making the switch permanent

Making the switch manager permanent

In the procedure Switching the master domain manager to the upgraded backup master *(on page 171)*, you switched your master domain manager promoting your new version backup master domain manager to the role of master domain manager.

To make this configuration fully operational and persistent through **JnextPlan**, you must complete the following procedure:

To make this configuration fully operational and persistent through **JnextPlan**, complete the following procedure on the new manager, referred to as *new_mgr_cpu*:

1. Edit the *localopts* file and modify the following entry as shown:

   ```
   DEFAULTWS=new_mgr_cpu
   ```

   where *new_mgr_cpu* is the workstation name of the new manager. For more information about `localopts` file, see Setting local options *(on page)*.
2. Change the workstation definition of the back-level master by running:

   ```
   composer modify cpu=old_mgr_cpu
   ```

   and in the definition substitute *type=manager* with *type=fta*
3. Change the workstation definition of the new manager by running:

   ```
   composer modify cpu=new_mgr_cpu
   ```

   and in the definition substitute *type=fta* with *type=manager*.
4. Ensure that the **optman** `cf` option is set to *all*.
5. Rebuild the plan to activate the changes to the database:

   ```
   JnextPlan -for 0000
   ```
6. Restore the previous setting of the **optman** `cf` option, if necessary.
7. Edit the `/TWA_DATA_DIR/mozart/globalopts` file and modify the **master=*old_mgr_cpu*** entry as shown:

   ```
   master=new_mgr_cpu
   ```

   where *new_mgr_cpu* is the workstation name of the new master. For more information about optman, see Setting global options *(on page)*.

   In this way the reports reptr-pre and reptr-post can run when you run **JnextPlan**.

Once you have made the switch manager permanent, you must run the FINAL job stream on the new master domain manager.

You can now proceed to Performing a direct upgrade of the master domain manager *(on page 172)*.

Performing a direct upgrade of the master domain manager

Performing a direct upgrade of the master domain manager

Upgrade a master domain manager by running the **serverinst** script.

1. Log in to the workstation where you plan to install.
2. On UNIX™ operating systems, ensure that **umask** is set to **022**. To verify that **umask** is set to the correct value, from a command prompt, run the **umask** command. If the value is different from **022**, modify it by running the following command:

   ```
   umask 022
   ```

   On Windows operating systems, ensure you have the correct rights on the folder where you plan to install.
3. Download the installation images from IBM Fix Central.
4. Browse to the folder `<image_location>/TWS/interp_name`.
5. Stop all IBM® Workload Scheduler services and WebSphere Application Server Liberty, by running the following commands:

   ```
   conman "stop;wait"
   conman "stopappserver;wait"
   conman "shut;wait"
   ShutDownLwa
   ```

6. If your current version is earlier than 10.2.1, your certificates need to be updated before they can work with version 10.2.3. The update is performed automatically, but you need to provide the password for the certificates. Define an environment variable with name **JKS_SSL_PASSWORD** and set it to the password you defined for the certificates. You can optionally encrypt the password using the secure script. For more information about the secure script, see Optional password encryption - secure script *(on page 300)*.
7. To update the database version, run the following command:

   On Windows operating systems

   ```
   cscript configureDb.vbs --rdbmstype db_type --dbhostname db_hostname --dbport db_port --
   dbname db_name --dbuser db_user --dbpassword db_password --dbadminuser db_administrator --
   dbadminuserpw db_administrator_password
   ```

   On UNIX operating systems

   ```
   ./configureDb.sh --rdbmstype db_type --dbhostname db_hostname --dbport db_port --dbname
   db_name --dbuser db_user --dbpassword db_password --dbadminuser db_administrator --
   dbadminuserpw db_administrator_password
   ```

   For more information about the configureDb script, see Database configuration - configureDb script *(on page 301)*.
8. Check your FIPS settings. Starting from version 10.2.1, FIPS is no longer supported. If FIPS is enabled in your current environment, specify the **enablefips** parameter when installing and set it to `false`. This will disable FIPS by changing the related keywords in the `localopts` and `ita.ini` files. For more information, see Q: My environment is FIPS compliant. What happens if I upgrade to version 10.2.3? *(on page 270)*. If FIPS is not enabled in your current environment, you can skip the **enablefips** parameter.
9. Start the installation launching the following command:

   On Windows operating systems

   ```
   cscript serverinst.vbs --acceptlicense yes --inst_dir INST_DIR
   ```

   On UNIX operating systems

   ```
   ./serverinst.sh --acceptlicense yes --inst_dir INST_DIR
   ```

   where *INST_DIR* is the directory where the component is installed. To find out the installation directory, see Finding out what has been installed in which IBM Workload Automation instances *(on page 25)*.

   For more information about the serverinst script, see Server components installation - serverinst script *(on page 310)*.

   Note: The **acceptlicense** and **inst_dir** parameters are required. You can also specify the following optional parameters:
   - **lang**
   - **work_dir**
   - **skipcheckprereq**
   - **enablefips**

   If you specify other parameters, they are ignored and the settings from the current instance are used instead.

10. After the installation has completed, run the following commands to start up IBM® Workload Scheduler services and WebSphere Application Server Liberty:

```
conman start
conman startappserver
StartUpLwa
```

11. To link all fault-tolerant agents, type the following command:

```
conman "link @!/@/@ "
```

12. If you have copied any template `.xml` files from the `templates` folder to the `overrides` folder, check for any differences between the default `.xml` files just upgraded in the `templates` folder and the files you are using in the `overrides` folder. If any differences are present, update the files in the `overrides` folder accordingly. For example, in version 10.2.3, the following variables have been added in the `ssl_config.xml` file and must be added in the corresponding file, if present in the `overrides` folder:

```
<jndiEntry id="keyStore.location" jndiName="keyStore.location" decode="false"
value="${server.config.dir}resources/security/${keyStore.location}"/>
<jndiEntry id="trustStore.location" jndiName="trustStore.location" decode="false"
value="${server.config.dir}resources/security/${trustStore.location}"/>
```

You have now successfully upgraded the master domain manager. You can now proceed to Switching back to the master domain manager from the backup master domain manager *(on page 174)*.

Switching back to the master domain manager from the backup master domain manager

After upgrading the old master domain manager to the 10.2.*x* version, you can now switch back the master capabilities, so that you restore your environment to the previous state, as follows:

1. Stop the application server as described in Application server - starting and stopping *(on page)*.
2. Switch the upgraded backup master domain manager, which now becomes the master domain manager, by issuing the following command from either the Dynamic Workload Console or the **command line** of your current backup master domain manager:

   From the Dynamic Workload Console

   In the navigation tree, click **Monitoring and Reporting > Monitor Workload >** select the engine and the object type Workstation, click run and, in the table of results, select backup master domain manager workstation name, click **More Actions**, and select **Become Master Domain Manager**.

   From the command line of the back-level master domain manager

   Issue the following command:

```
conman "switchmgr masterdm;new_mgr_cpu"
```

   where *new_mgr_cpu* is the backup master domain manager workstation name.
3. Switch the event processor from the old backup master domain manager to the master domain manager, by running the following command from either the Dynamic Workload Console or the **command line** of your old backup master domain manager:
4. Restart the application server as described in Application server - starting and stopping *(on page)*.

You have now successfully switched back the upgraded master domain manager. You can now proceed to Making the switch permanent *(on page 175)*.

Making the switch permanent

Making the switch manager permanent

In the procedure Switching the master domain manager to the upgraded backup master *(on page 171)*, you switched your master domain manager promoting your new version backup master domain manager to the role of master domain manager.

To make this configuration fully operational and persistent through **JnextPlan**, you must complete the following procedure:

To make this configuration fully operational and persistent through **JnextPlan**, complete the following procedure on the new manager, referred to as *new_mgr_cpu*:

1. Edit the *localopts* file and modify the following entry as shown:

   ```
   DEFAULTWS=new_mgr_cpu
   ```

   where *new_mgr_cpu* is the workstation name of the new manager. For more information about `localopts` file, see Setting local options *(on page)*.
2. Change the workstation definition of the back-level master by running:

   ```
   composer modify cpu=old_mgr_cpu
   ```

   and in the definition substitute *type=manager* with *type=fta*
3. Change the workstation definition of the new manager by running:

   ```
   composer modify cpu=new_mgr_cpu
   ```

   and in the definition substitute *type=fta* with *type=manager*.
4. Ensure that the **optman** `cf` option is set to *all*.
5. Rebuild the plan to activate the changes to the database:

   ```
   JnextPlan -for 0000
   ```

6. Restore the previous setting of the **optman** `cf` option, if necessary.
7. Edit the */TWA_DATA_DIR*`/mozart/globalopts` file and modify the **master=*old_mgr_cpu*** entry as shown:

   ```
   master=new_mgr_cpu
   ```

   where *new_mgr_cpu* is the workstation name of the new master. For more information about optman, see Setting global options *(on page)*.

   In this way the reports reptr-pre and reptr-post can run when you run **JnextPlan**.

Once you have made the switch manager permanent, you must run the FINAL job stream on the new master domain manager.

You can now proceed to Upgrading agents and domain managers *(on page 175)*.

Upgrading agents and domain managers

There are several methods you can choose from to upgrade your domain managers and agents.

The agent upgrade can be performed with minimal impact to scheduling activities. The agents are stopped for the shortest time necessary to perform the maintenance. Any active agent command-line interfaces and processes, such as conman, composer, netman, mailman, and batchman, to name a few, continue running. Any jobs already running when the upgrade process begins, continue to run as planned, however, no new jobs begin execution during this time. Once the upgrade is complete, the agent is restarted and quickly reconnects with its jobs. Any jobs that were actively running before the upgrade that have not yet completed, continue to run, and any jobs that successfully finished running during the upgrade procedure report a successful job status. An automatic backup and restore feature is in place in case of failure.

Because domain managers are agents, they are upgraded using the procedures described in this section.

If you choose to upgrade your environment top-down, then the agents get upgraded progressively after you have upgraded the master domain manager and its backup. This means that new features and enhancements are not available on all of your agents at the same time. If, instead, you choose to upgrade your environment bottom-up, then the agents are upgraded first, and new features and enhancements become available after the master domain manager and its backup have been upgraded.

Important: After upgrading your fault-tolerant agents, it might be necessary to manually update the security file on the fault-tolerant agents in your environment to add access to folders for all of the scheduling objects that can be defined or moved into folders. These updates are especially important if you plan to use the command line on the fault-tolerant agents to perform operations on the objects in folders. See Updating the security file *(on page)* for more information.

You can choose to upgrade your agents using any of the following methods:

twsinst script

A single line command that checks if processes or a command line is running before it starts. It saves disk space and RAM because it is not Java-based. See Upgrade procedure *(on page 201)* and Upgrading agents on IBM i systems *(on page 205)*

Centralized agent update

Upgrade or update multiple fault-tolerant agent and dynamic agent instances at the same time. Download the fix pack installation package, or the eImage upgrade package to the master domain manager and then either run the installation on multiple agent instances or schedule the installation by creating and submitting a job to run. This upgrade method is not supported on z-centric agent instances. See Centralized agent update *(on page 208)*.

HCL BigFix

Upgrade IBM® Workload Scheduler agents using HCL BigFix analyses and fixlets. You can choose to schedule the upgrade or you can run it immediately. See Upgrading agents using HCL BigFix *(on page 216)*.

For a list of supported operating systems and requirements, see the System Requirements Document at IBM Workload Scheduler Detailed System Requirements.

When the upgrade procedure has completed successfully, the backup instance is deleted.

Note: The `localopts` file is not modified during the agent upgrade process. The file generated by the upgrade process is saved to the `/config` directory to maintain your custom values, if any. You can then merge the two files with your customized values and save the resulting file in the following path:

On Windows operating systems

<TWA_home>`\TWS`

On UNIX operating systems

<TWA_DATA_DIR>

When upgrading dynamic agents featuring both a local and a remote gateway, ensure you either upgrade the agent first and then the gateway or upgrade both at the same time.

Parallel upgrade from version 9.5.0.*x* or 10.*x*.*x* to version 10.2.3

To upgrade your environment using a parallel upgrade procedure, perform the following steps:

1. Converting default certificates *(on page 178)*, if you are using default certificates in your current environment. Use this procedure to convert the certificates from the JKS to the PEM format, then copy them to the workstations where you plan to install the server components (dynamic domain manager and its backups, master domain manager and its backups) and the Dynamic Workload Console.

   Certificates are now required when installing or upgrading IBM® Workload Scheduler. You can no longer install nor upgrade IBM® Workload Scheduler without securing your environment with certificates. The required certificates are:
   - ca.crt
   - tls.key
   - tls.crt

   For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).
2. Upgrading WebSphere Application Server Liberty *(on page 179)*
3. Encrypting passwords (optional) *(on page 181)*
4. Upgrading the Dynamic Workload Console and its database *(on page 182)*
5. Creating the IBM Workload Scheduler administrative user *(on page 184)* on the workstations which will host the components at 10.2.3 level.
6. Upgrading the database for the server components *(on page 184)*
7. Installing a new dynamic domain manager configured as a backup *(on page 187)*
   a. Switching the manager to the upgraded backup *(on page 189)*
   b. Making the switch permanent *(on page 189)*
8. Installing the new master domain manager configured as a backup *(on page 190)*
   a. Switching the manager to the upgraded backup *(on page 194)*
   b. Making the switch permanent *(on page 195)*
9. Customizing and submitting the optional FINAL job stream *(on page 195)*
10. Installing a new backup dynamic domain manager *(on page 197)* to replace the backup dynamic domain manager which you have switched to become the current dynamic domain manager.
11. Cleaning up your environment *(on page 199)*
12. Optionally dismiss all back-level components
13. Upgrading agents and domain managers *(on page 200)*
14. Optionally install a new backup master domain manager at version 10.2.3 to ensure failover capabilities.

Environment with custom certificates

If you have version 9.5 installed with custom certificates, then after upgrading to 10.2 you must ensure that the parameters and the name of the relevant certificates in the **localopts** file are correct.

If you have previously used certificates generated with OpenSSL, check the paths in the following section:

- For Open SSL, check:
  - SSL key
  - SSL certified
  - SSL key pwd
  - SSL CA certified
  - SSL random seed

If you have used GSKit, the relevant parameters are automatically migrated to the new OpenSSL parameters:

- **SSL Version**
- **SSL Ciphers**
- **CLI SSL Ciphers**
- **CLI SSL Version**

Converting default certificates

Procedure to extract and convert default certificates generated in your current version prior to upgrading.

If you are using default certificates, extract and convert them before you start the upgrade. Perform the following steps:

1. Set the IBM® Workload Scheduler environment, as described in Setting the environment variables *(on page 138)*.
2. To ensure the keytool and openssl commands start correctly on all operating systems, browse to the folder where the keytool and openssl commands are located and launch the commands as follows:

   ```
   cd <TWS_DIR>/JavaExt/jre/jre/bin
   ```

   ```
   ./keytool -importkeystore -srckeystore TWSServerKeyFile.jks -destkeystore
   <path_of_extracted_certs>/server.p12 -deststoretype pkcs12
   ```

   ```
   cd <TWS_DIR>/tmpOpenSSL64/1.1/bin/openssl
   ```

   ```
   ./openssl pkcs12 -in <path_of_extracted_certs>/server.p12 -out
   <path_of_extracted_certs>/tls.tot
   ```

   The location of the `TWSServerKeyFile.jks` varies depending on the IBM® Workload Scheduler version you have currently installed, as follows:

   versions 9.5 and later

   TWA_DATA_DIR`/usr/servers/engineServer/resources/security`

   versions 9.4 and earlier

   TWA_home`/WAS/TWSProfile/etc`

3. Open the `tls.tot` file with any text editor.
4. From the `tls.tot` file, copy the private key to a new file named `tls.key`.
   The `tls.key` file must be structured as follows:

   ```
   ----BEGIN ENCRYPTED PRIVATE KEY----
   <private_key>
   ----END ENCRYPTED PRIVATE KEY----
   ```

   Note: Insert a carriage return after each key, so that an empty line is inserted after each key.

5. From the `tls.tot` file, copy the public key to a new file named `tls.crt`.
   The `tls.crt` file must be structured as follows:

   ```
   ----BEGIN CERTIFICATE----
   <public_key>
   ----END CERTIFICATE----
   ```

   Note: Insert a carriage return after each key, so that an empty line is inserted after each key.

6. Copy the contents of the `tls.crt` file into a new file named `ca.crt`. If you want to upgrade a dynamic domain manager, also copy the contents of the `tls.crt` file into another new file named `jwt.crt`.
7. Create a file named `tls.sth` containing the passphrase you have specified for creating the `.p12` certificate in step 2 *(on page 178)*, encoded in `base64` format. To create the `tls.sth` file, use the following command:

   ```
   ./secure -password your_password -base64 e -out
   <path_of_extracted_certs>/tls.sth
   ```

   If you are using a version earlier than 10.*x*, you can find the secure script in the installation package of the 10.2.3 version you are upgrading to. You can launch the script from on of the following paths:

   master domain manager and agent

   ```
   <10.2.3_extracted_image_dir>/TWS/<interp>/Tivoli_LWA_<interp>/TWS/bin
   ```

   Dynamic Workload Console

   ```
   <10.2.3_extracted_image_dir>/DWC/<interp>/bin
   ```

   where

   <interp>

   is the operating system you are installing on

   As an alternative, you can use the following command on UNIX workstations:

   ```
   echo -n "passwordToEncode" | base64 >> tls.sth
   ```

8. Browse to the GSKit folder and extract the client certificates from the *TWA_DATA_DIR*`/ssl/GSKit` folder by running the following commands, depending on the IBM® Workload Scheduler version you have currently installed:

   ```
   cd <TWS_DIR>/tmpGSKit64/8/bin
   ```

   versions 9.5 and later

   ```
   ./gsk8capicmd_64 -cert -extract -db <TWA_DATA_DIR>/ssl/GSKit/TWSClientKeyStore.kdb
   -stashed -label client -target <path_of_extracted_certs>/client.crt
   ```

   versions 9.4 and earlier

   ```
   ./gsk8capicmd_64 -cert -extract -db <TWS_DIR>/ssl/GSKit/TWSClientKeyStore.kdb
   -stashed -label client -target <path_of_extracted_certs>/client.crt
   ```

9. Create a folder named `additionalCAs` in the folder where you extracted the certificates and move the `client.crt` file created in step 8 *(on page 179)* to the `additionalCAs` folder.
10. Insert the `client.crt` in the `additionalCAs` folder when providing the certificates to the installation script with the **sslkeysfolder** parameter.
11. Assign the correct permissions (755) and ownerships to extracted certificates, as follows:

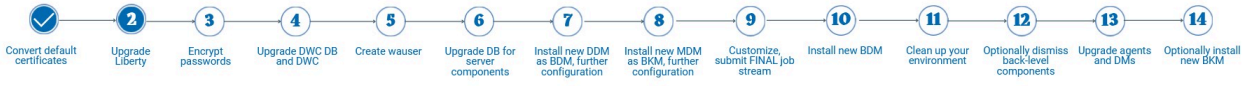
    ```
    chmod -R 755 <path_of_extracted_certs>
    ```

You have now extracted and converted your certificates for use with version 10.2.3.

You can now upgrade WebSphere Application Server Liberty, as described in Upgrading WebSphere Application Server Liberty *(on page 179)*. When upgrading IBM® Workload Scheduler components in upcoming steps, provide the path to the folder where you extracted the certificates using the **sslkeysfolder** parameter when running the installation scripts. For more information about the installation scripts, see Reference *(on page 300)*.

Upgrading WebSphere Application Server Liberty

Upgrading WebSphere Application Server Liberty to the latest supported version. This is an optional step you might want to perform before you upgrade the Dynamic Workload Console and the server components.



On AIX and Linux workstations, ensure you permanently set the **ulimit** parameter as follows:

- data segment process (option **-d**) = `unlimited`
- file size (option **-f**) = `unlimited`
- max user processes (option **-u**) = `>260000` up to `unlimited`
- open files (option **-n**) = `>100000` up to `unlimited`
- max memory size (option **-m**) = `unlimited`
- stack size (option **-s**) = `>33000` up to `unlimited`

On the master domain manager, these settings must be applied to:

- `root`
- the IBM® Workload Scheduler administrative user

On the Dynamic Workload Console, these settings must be applied to:

- `root`
- the Dynamic Workload Console installation user (if this user is different from root)

Ensure that your system meets the operating system and Java requirements. For more information, see WebSphere Application Server Liberty Base detailed system requirements.

You can quickly install WebSphere Application Server Liberty Base by extracting an archive file on all supported platforms.

Install WebSphere Application Server Liberty Base on all of the following workstations, which comprise a typical installation:

- master domain manager
- backup domain manager
- two Dynamic Workload Console installations on two separate workstations

If you plan to install a dynamic domain manager and its backup, these components require a separate WebSphere Application Server Liberty Base installation.

To extract the archive, you can use your own Java Ext or use the Java Ext provided with the IBM® Workload Scheduler image. The provided Java Ext is located in the following path in the image for your operating system: `<IMAGE_DIR>/TWS/<INTERP>/Tivoli_Eclipse_<INTERP>/TWS/JavaExt/`.

To install WebSphere Application Server Liberty Base, perform the following steps:

1. Find out which version of WebSphere Application Server Liberty Base is required, by running the Detailed Software Requirements report and browsing to the **Prerequisites** tab.
2. Download WebSphere Application Server Liberty Base from Recommended updates for WebSphere Application Server Liberty.
3. Install WebSphere Application Server Liberty Base by extracting the archive file to a directory of your choice.

   On Windows operating systems

   ```
   java -jar <liberty_download_dir>\wlp-base-all-<version>.jar
   --acceptLicense <install_dir>
   ```

   On UNIX operating systems

   ```
   ./java -jar <liberty_download_dir>/wlp-base-all-<version>.jar
   --acceptLicense <install_dir>
   ```

   where:

   <liberty_download_dir>

   The directory where you downloaded WebSphere Application Server Liberty Base.

   <version>

   The number of the version.

   <install_dir>

The directory where you want to install WebSphere Application Server Liberty Base.

Note: Note that the value of the *<install_dir>* parameter must match the value to be defined for the **wlpdir** parameter when installing the master domain manager and its backup, dynamic domain manager and its backup, and the Dynamic Workload Console.

4. Ensure the IBM® Workload Scheduler administrative user has the rights to run WebSphere Application Server Liberty Base and full access to the installation directory. If WebSphere Application Server Liberty Base is shared between the master domain manager and the Dynamic Workload Console, ensure also the Dynamic Workload Console user has the same rights.

You have now successfully installed WebSphere Application Server Liberty Base.

You can now proceed to Encrypting passwords (optional) *(on page 181)* or to Upgrading the Dynamic Workload Console and its database *(on page 182)*.

Encrypting passwords (optional)

How to encrypt passwords required by the installation process

You can optionally encrypt the passwords that you will use while installing, upgrading, and managing IBM® Workload Scheduler. The secure command uses the AES method and prints the encrypted password to the screen or saves it to a file.

Note: It is important you understand the limits to the protection that this method provides. The custom passphrase you use to encrypt the passwords is stored in clear format in the `passphrase_variables.xml` file, stored in `configureDropin`. To fully understand the implications of this method, it is recommended you read the information provided by WebSphere Application Server Liberty Base at the link Liberty: The limits to protection through password encryption.

You can perform a typical procedure, which uses a custom passphrase, as described in the following scenario. For more information about all secure arguments and default values, see Optional password encryption - secure script *(on page 300)*.

Encrytping the password

1. Browse to the folder where the secure command is located:
   - Before the installation, the command is located in the product image directory, `<image_directory>/TWS/<op_sys>/Tivoli_LWA_<op_sys>/TWS/bin`
   - After the installation, the command is located in `TWA_home/TWS/bin`
2. Depending on your operating system, encrypt the password as follows:

   Windows operating systems

   ```
   secure -password password -passphrase passphrase
   ```

   UNIX operating systems

   ```
   ./secure -password password -passphrase passphrase
   ```

   z/OS operating systems

   ```
   ./secure -password password -passphrase passphrase
   ```

   where

   -password

Specifies the password to be encrypted.

-passphrase

Optional. Specifies the custom passphrase that is used to generate the key with which the command encrypts the password. If you set this parameter, inform the user who installs IBM Workload Automation that they must define the **SECUREWRAP_PASSPHRASE** environment variable in the same shell from which they run the installation command, and set it to the same value as the **passphrase** argument. On Windows operating systems, the passphrase must be at least 8 characters long.

3. Provide both the encrypted password and custom passphrase to the user in charge of installing IBM Workload Automation. You can use encrypted passwords only in association with the specific passphrase used to encrypt them.

Installing with the encrypted password

The user in charge of installing IBM Workload Automation must set the **SECUREWRAP_PASSPHRASE** environment variable by performing the following steps:

1. Open a brand new shell session.
2. Ensure that no value is set for the **SECUREWRAP_PASSPHRASE** environment variable.
3. Define the **SECUREWRAP_PASSPHRASE** environment variable and set it to the passphrase defined by the user who ran the secure command, as follows:

```
SECUREWRAP_PASSPHRASE=<passphrase>
```

You can use encrypted passwords only in association with the specific passphrase used to encrypt them.

4. In the same shell session, provide the encrypted passwords when running any command that uses a password. An encrypted password looks like the following example:

```
{aes}AFC3jj9cROYyqR+3CONBzVi8deLb2Bossb9GGroh8UmDPGikIkzXZzid3nzY0IhnSg=
```

You can now proceed to Upgrading the Dynamic Workload Console and its database *(on page 182)*.

Upgrading the Dynamic Workload Console and its database

Upgrade the Dynamic Workload Console from version 9.5.0.*x* or 10.*x* to version 10.2.*x*. If you have several Dynamic Workload Console nodes in a cluster, upgrade all the nodes in the cluster.

When upgrading the IBM Workload Scheduler environment, it is a good practice to update the Dynamic Workload Console first. If you update the console, you can then use it to verify that your environment is working after updating the remaining components.

Note: If you are installing the Dynamic Workload Console version 10.2.3 or later, the Federator is also automatically installed. This component enables you to monitor your objects through the Orchestration Monitor page of the Dynamic Workload Console. For detailed information about how to configure and use the Federator, see Mirroring the z/OS current plan to enable the Orchestration Monitor *(on page)*.

If you are currently using Derby, you need to install a supported database and migrate your data. This is necessary because Derby is no longer supported as of version 10.2.3. For more information, see Connecting the Dynamic Workload Console to a new node or database *(on page 161)*.

1. Log in to the workstation where you plan to install the Dynamic Workload Console.
2. On UNIX™ operating systems, ensure that **umask** is set to **022**. To verify that **umask** is set to the correct value, from a command prompt, run the **umask** command. If the value is different from **022**, modify it by running the following command:

```
umask 022
```

On Windows operating systems, ensure you have the correct rights on the folder where you plan to install.

3. Download the installation images from IBM Fix Central.
4. Browse to the folder *image_location*.
5. If possible, stop all Dynamic Workload Console instances.
   If this is not possible, launch the configureDB script at a time when the Dynamic Workload Console is processing a low workload. If the configureDB script should fail because of conflicts with the Dynamic Workload Console, restart the script.
6. If your current version is earlier than 10.2.1, your certificates need to be updated before they can work with version 10.2.3. The update is performed automatically, but you need to provide the password for the certificates. Define an environment variable with name **JKS_SSL_PASSWORD** and set it to the password you defined for the certificates. You can optionally encrypt the password using the secure script. For more information about the secure script, see Optional password encryption - secure script *(on page 300)*.
7. To update the database version, run the following command:

   On Windows operating systems

   ```
   cscript configureDb.vbs --rdbmstype db_type --dbhostname db_hostname
   --dbport db_port --dbname db_name --dbuser db_user
   --dbpassword db_password --dbadminuser db_administrator
   --dbadminuserpw db_administrator_password
   ```

   On UNIX operating systems

   ```
   ./configureDb.sh --rdbmstype db_type --dbhostname db_hostname
   --dbport db_port --dbname db_name --dbuser db_user
   --dbpassword db_password --dbadminuser db_administrator
   --dbadminuserpw db_administrator_password
   ```

   On z/OS operating systems

   ```
   ./configureDb.sh --rdbmstype db_type --dbhostname DB_hostname
   --dbport db_port --dbname db_name --dbuser db_user
   --dbadminuser DB_admin_user --dbadminuserpw DB_admin_pwd
   --zlocationname zOS_location_containing_db --zbufferpoolname buffer_pool_in_zOS_location
   ```

   For more information about all parameters and supported values of the `configureDb` command, see Database configuration - configureDb script *(on page 301)*.
8. Start the upgrade by launching the following command:

   On Windows operating systems

   ```
   cscript dwcinst.vbs --acceptlicense yes --inst_dir INST_DIR
   ```

   On UNIX operating systems

   ```
   ./dwcinst.sh --acceptlicense yes --inst_dir INST_DIR
   ```

   On z/OS operating systems

   ```
   ./dwcinst.sh --acceptlicense yes --inst_dir INST_DIR
   ```

   For further details about commands, see Reference *(on page 300)*.
9. If you had previously exported the Dynamic Workload Console, as described in Connecting the Dynamic Workload Console to a new node or database *(on page 161)*, you can now import them in the new Dynamic Workload Console from the **Administration** > **Manage Settings** menu. If you have a high availability configuration, import the settings on one node.
10. If you have copied any template `.xml` files from the `templates` folder to the `overrides` folder, check for any differences between the default `.xml` files just upgraded in the `templates` folder and the files you are using in the `overrides` folder. If any differences are present, update the files in the `overrides` folder accordingly. For example, in version 10.2.3, the following variables have been added in the `ssl_config.xml` file and must be added in the corresponding file, if present in the `overrides` folder:

    ```
    <jndiEntry id="keyStore.location" jndiName="keyStore.location" decode="false"
    value="${server.config.dir}resources/security/${keyStore.location}"/>
    <jndiEntry id="trustStore.location" jndiName="trustStore.location" decode="false"
    value="${server.config.dir}resources/security/${trustStore.location}"/>
    ```

You have now successfully upgraded the Dynamic Workload Console. You can now proceed to Creating the IBM Workload Scheduler administrative user *(on page 184)*.

Creating the IBM® Workload Scheduler administrative user

Instructions to create the IBM® Workload Scheduler administrative user.

IBM® Workload Scheduler administrative user

The IBM® Workload Scheduler administrator creates the administrative user (**wauser**). The administrative user is the user for which the product will be installed in the subsequent steps. This implies that this user has full access to all scheduling objects.

The user name can contain alphanumeric, dash (-), and underscore (_) characters; it cannot contain national characters. The first character of the user name must be a letter.

The following considerations apply:

On Windows operating systems:

- If this user account does not already exist, it is automatically created at installation time.
- If installing on a Windows™ server in a domain, do not define a domain and local ID with the same user name.
- If you specify a domain user, define the name as *domain_name\user_name*.
- If you specify a local user, define the name as *system_name\user_name*. Type and confirm the password.

On UNIX and Linux operating systems:

This user account must be created manually before running the installation and must be enabled to login to the machine where the master domain manager is going to be installed. Create a user with a home directory and group. Use the appropriate UNIX and Linux operating system commands to create the user.

Important: Group names that contain a "/" (forward slash) character can cause permissions to not be set correctly. When IBM® Workload Scheduler retrieves credentials from WebSphere Application Server Liberty, it parses the returned list of groups names assuming they are saved in the format `<realm_name>/<group_name>`. If the group name, the realm name, or both contain a "/" character, the parsing fails.

You can also install IBM® Workload Scheduler using a user different from the root user. This installation method is known as **no-root installation** and applies to all IBM® Workload Scheduler components. Note that if you choose this installation method, only the user who performs the installation can use IBM® Workload Scheduler. For this reason, the typical installation scenario described in this section uses the root user.

For more information, see IBM Workload Scheduler user management *(on page 34)*.

What to do next

You can now proceed to Upgrading the database for the server components *(on page 184)*.

Upgrading the database for the server components

Upgrade the database tables before upgrading the server components.

Note: Before upgrading the database schema, ensure you have created a backup. Refer to the documentation related to your RDBMS for information about the backup procedure.

Ensure you have acquired information about the IBM® Workload Scheduler tablespaces that were specified when the database tables were created and populated the first time. If values different from the default values were used, then your database administrator must provide them for this upgrade procedure. If default values were used, then they do not need to be specified during the upgrade procedure. The default values for the IBM® Workload Scheduler data, log, and plan tablespaces are as follows:

- **--iwstsname** `TWS_DATA`
  - For Oracle only, the default is `USERS`
- **--iwslogtsname** `TWS_LOG`
  - For Oracle only, the default is `USERS`
- **--iwsplantsname** `TWS_PLAN`
  - For Oracle only, the default is `USERS`

For more information about all parameters and supported values of the `configureDb` command, see Database configuration - configureDb script *(on page 301)*.

You can run the configureDb command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

The script creates an SQL file with all the statements needed to upgrade the IBM® Workload Scheduler database schema to the latest version and, by default, automatically applies it.

Default values are stored in the `configureDb<`*`database_vendor`*`>.properties` file, located in *`image_location`*`/TWS/`*`interp_name`*. For an example of a properties file, see What is the content of a database properties file? *(on page 66)*.

If you need to modify any of the default values, edit the `configureDb<`*`database_vendor`*`>.properties` file, but do not modify the `configureDb<`*`database_vendor`*`>.template` file located in the same path.

To upgrade the IBM® Workload Scheduler database schema, perform the following steps:

1. On the workstation where you plan to install the new backup master domain manager or backup dynamic domain manager, extract the IBM® Workload Scheduler package at the latest version to a directory of your choice.
2. Browse to the *`image_location`*`/TWS/`*`interp_name`* path.
3. Type the following command to upgrade the IBM® Workload Scheduler database schema to the latest version. Ensure that you use the same database administrator credentials you used when the IBM® Workload Scheduler database schema objects were created. The new backup master domain manager or backup dynamic domain manager is configured to point to the existing database instance.

   On Windows operating systems

   ```
   cscript configureDb.vbs --rdbmstype db_vendor --dbhostname db_hostname --dbport db_port
   --dbname db_name --dbuser db_user  --componenttype server_component
   --dbadminuser db_administrator --dbadminuserpw db_administrator_password
   --iwstsname tablespace_data --iwslogtsname tablespace_log --iwsplantsname tablespace_plan
   ```

   On UNIX operating systems

   ```
   ./configureDb.sh --rdbmstype db_vendor --dbhostname db_hostname --dbport db_port
   --dbname db_name --dbuser db_user  --componenttype server_component
   --dbadminuser db_administrator --dbadminuserpw db_administrator_password
   --iwstsname tablespace_data --iwslogtsname tablespace_log --iwsplantsname tablespace_plan
   ```

   where:

   --rdbmstype

   The database vendor.

   --dbhostname *db_hostname*

   The host name or IP address of database server.

--dbport ***db_port***

The port of the database server.

--dbname ***db_name***

The name of the IBM® Workload Scheduler database.

--dbuser ***db_user***

The user that has been granted access to the IBM® Workload Scheduler tables on the database server.

--dbpassword ***db_password***

The password for the user that has been granted access to the IBM® Workload Scheduler tables on the database server. Special characters are not supported.

--dbadminuser ***db_admin_user***

The database administrator user that creates the IBM® Workload Scheduler schema objects on the database server.

--dbadminuserpw ***db_admin_password***

The password of the DB administrator user that creates the IBM® Workload Scheduler schema objects on the database server. Special characters are not supported.

--componenttype MDM | DDM

The IBM® Workload Scheduler component for which the database is installed. This parameter is optional. Supported values are:

MDM

master domain manager.

DDM

dynamic domain manager.

--iwstsname ***tablespace_data***

The name of the tablespace for IBM® Workload Scheduler data. The default value for all supported RDBMS is TWS_DATA, with the exception of Oracle where the default is USERS.

--iwslogtsname ***tablespace_log***

The name of the tablespace for the IBM® Workload Scheduler log. The default value for all supported RDBMS is TWS_LOG, with the exception of Oracle where the default is USERS.

--iwsplantsname ***db_port***

The name of the tablespace for the IBM® Workload Scheduler plan. The default value for all supported RDBMS is TWS_PLAN, with the exception of Oracle where the default is USERS.

--auth_type ***db_name***

The MSSQL authentication mode. The default is SQLSERVER which uses native SQL authentication.

You can optionally point the backup master domain manager to different database residing on the sane workstation. For more information, see Connecting the master domain manager to a new database *(on page)*.

Note: The following parameters specified with the configureDb command are also required when you upgrade the server components with the serverinst command and their values must be the same:

- **rdbmstype**
- **dbhostname**
- **dbport**
- **dbname**
- **dbuser**
- **dbpassword**

You have now successfully upgraded the database schema for the IBM® Workload Scheduler database.

You can now proceed to Installing a new dynamic domain manager configured as a backup *(on page 187)* or to Installing the new master domain manager configured as a backup *(on page 190)*.

Installing a new dynamic domain manager configured as a backup

Procedure for installing a dynamic domain manager configured as a backup

Install a new dynamic domain manager at the latest product version level configured as the new backup dynamic domain manager by running the serverinst script.

The procedure to install the dynamic domain manager and backup dynamic domain manager is exactly the same, with the difference that it is performed on two different workstations and that each installation points to its local WebSphere Application Server Liberty Base installation. IBM® Workload Scheduler determines whether or not a dynamic domain manager is already present in the environment and proceeds to install a dynamic domain manager or backup dynamic domain manager accordingly.

The IBM® Workload Scheduler administrator installs the dynamic domain manager as the backup. The following information is required:

Table 13. Required information

Required information for performing the installation

Command parameter	Information type	Provided in...
Database information		
--rdbmstype	database type	Upgrading the database for the server components *(on page 239)*
--dbhostname	database hostname	
--dbport	database port	
--dbname	database name	
--dbuser	database user name	
--dbpassword	database password	
IBM® Workload Scheduler information		
--wauser	IBM® Workload Scheduler administrative user name	Creating the IBM Workload Scheduler administrative user *(on page 239)*
--wapassword	IBM® Workload Scheduler administrative user password	
WebSphere Application Server Liberty Base information		
--wlpdir	WebSphere Application Server Liberty Base installation directory	Installing WebSphere Application Server Liberty *(on page 225)*
Security information		
--sslkeysfolder	location of converted certificates	Converting default certificates *(on page 223)*
--sslpassword	password of converted certificates	Converting default certificates *(on page 223)*

Before starting the installation, ensure the following steps have been completed:

1. Converting default certificates *(on page 178)*. Because you are installing a dynamic domain manager, also copy locally the `jwt.crt` file created in the conversion procedure.
2. Upgrading WebSphere Application Server Liberty *(on page 179)* on the workstation where you plan to install the dynamic domain manager and on the workstation where you plan to install the backup dynamic domain manager.
3. Encrypting passwords (optional) *(on page 181)*
4. Upgrading the Dynamic Workload Console and its database *(on page 182)*
5. Creating the IBM Workload Scheduler administrative user *(on page 184)* on the workstations which will host the components at 10.2.3 level.
6. Upgrading the database for the server components *(on page 184)*

You can run the **serverinst** command specifying a typical set of parameters. In this case, default values are used for all remaining parameters. For more information about all **serverinst** parameters and default values, see Server components installation - serverinst script *(on page 310)*.

A properties file named `serverinst.properties` is available if you do not want to type parameters in the command line. This is expecially useful if you need to specify many parameters or if you want to reuse the file for several installations. The file is located in *image_location*`/TWS/`*interp_name*.

You can specify values in the properties file, type them in the command line, or use both methods. If a parameter is specified both in the properties file and in the command line, the command line value takes precedence.

If you need to modify any of the default values, edit the `serverinst.properties` file, but do not modify the `serverinst.template` file located in the same path.

To install the dynamic domain manager as a backup, perform the following steps:

1. Log in to the workstation where you plan to install.
2. Browse to the folder where the `serverinst` command is located:

   On Windows operating systems

   image_location`\TWS\`*interp_name*

   On UNIX operating systems

   image_location`/TWS/`*interp_name*

3. Start the installation specifying a typical set of parameters. In this case, default values are used for all remaining parameters:

   On Windows operating systems

   ```
   cscript serverinst.vbs --acceptlicense yes --rdbmstype db_type
           --dbhostname db_hostname --dbport db_port --dbname db_name
           --dbuser db_user --dbpassword db_password --wauser wa_user
           --wapassword wa_password --componenttype DDM --domain domain_name
           --master mdm_name --mdmbrokerhostname mdm_broker_host_name
           --sslkeysfolder certificate_files_path --sslpassword keystore_truststore_password
           --mdmhttpsport mdm_https_host_name --wlpdir Liberty_installation_dir\wlp
   ```

   On UNIX operating systems

   ```
   ./serverinst.sh --acceptlicense yes --rdbmstype db_type
           --dbhostname db_hostname --dbport db_port --dbname db_name
           --dbuser db_user --dbpassword db_password --wauser wa_user
           --wapassword wa_password --componenttype DDM --domain domain_name
           --master mdm_name --mdmbrokerhostname mdm_broker_host_name
           --sslkeysfolder certificate_files_path --sslpassword keystore_truststore_password
           --mdmhttpsport mdm_https_host_name --wlpdir Liberty_installation_dir/wlp
   ```

4. Distribute the Symphony file to the new dynamic domain manager configured as backup:
   a. Ensure that the **optman cf** option is set to *all*.
   b. To distribute the Symphony file to the new dynamic domain manager configured as backup, run JnextPlan -for 0000 or wait until the end of the production plan.
   c. Restore the previous setting of the **optman cf** option, if you previously modified the value.

You have now successfully installed the backup dynamic domain manager at the new product version level.

For more information about all **serverinst** parameters and default values, see Server components installation - serverinst script *(on page 310)*.

What to do next

You can now proceed to Switching the manager to the upgraded backup *(on page 189)*.

Switching the manager to the upgraded backup

To switch the back-level manager to the upgraded backup, complete the following procedure:

1. Switch to your upgraded backup manager, which now becomes your current active manager, by issuing the following command from either the Dynamic Workload Console or the **command line** of your back-level manager:

   From the Dynamic Workload Console

   In the navigation tree, click **Monitoring and Reporting > Monitor Workload >** select the engine and the object type Workstation, click **Run** and, in the table of results, select backup manager workstation name, click **More Actions**, and select **Become Master Domain Manager**or **Become Dynamic Domain Manager**, as necessary.

   From the command line of the back-level manager

   Issue the following command:

   ```
   conman "switchmgr masterdm;new_mgr_cpu"
   ```

   where *new_mgr_cpu* is the backup manager workstation name.

2. Switch the event processor from the back-level manager to the backup manager, by running the following command from either the Dynamic Workload Console or the **command line** of your back-level manager:

   From the Dynamic Workload Console

   In the navigation tree, click **Monitoring and Reporting > Monitor Workload >** select the engine and the object type Workstation, click run and, in the table of results, select manager workstation name, click **More Actions**, and select **Become Event Processor**.

   From the command line of the back-level manager

   Issue the following command:

   ```
   conman "switcheventprocessor new_mgr_cpu"
   ```

   where *new_mgr_cpu* is the backup manager workstation name.

You have now successfully switched back-level manager to the upgraded backup.

You can now proceed to make this switch permanent, as described in Making the switch permanent *(on page 189)*.

Making the switch permanent

Making the switch permanent (DDM)

In the procedure Switching the manager to the upgraded backup *(on page 189)*, you switched your manager promoting your new version backup manager to the role of manager.

To make this configuration fully operational and persistent through **JnextPlan**, complete the following procedure on the new manager, referred to as *new_mgr_cpu*:

1. Edit the *localopts* file and modify the following entry as shown:

   ```
   DEFAULTWS=new_mgr_cpu
   ```

   where *new_mgr_cpu* is the workstation name of the new manager. For more information about `localopts` file, see Setting local options *(on page)*.
2. Change the workstation definition of the back-level master by running:

   ```
   composer modify cpu=old_mgr_cpu
   ```

   and in the definition substitute *type=manager* with *type=fta*
3. Change the workstation definition of the new manager by running:

   ```
   composer modify cpu=new_mgr_cpu
   ```

   and in the definition substitute *type=fta* with *type=manager*.
4. Ensure that the **optman** `cf` option is set to *all*.
5. Rebuild the plan to activate the changes to the database:

   ```
   JnextPlan -for 0000
   ```

6. Restore the previous setting of the **optman** `cf` option, if necessary.
7. Edit the */TWA_DATA_DIR*`/mozart/globalopts` file and modify the **master=*old_mgr_cpu*** entry as shown:

   ```
   master=new_mgr_cpu
   ```

   where *new_mgr_cpu* is the workstation name of the new master. For more information about optman, see Setting global options *(on page)*.

   In this way the reports reptr-pre and reptr-post can run when you run **JnextPlan**.

You have now successfully made the switch permanent.

You have now to import to the new dynamic domain manager the security file from the previous dynamic domain manager, as follows:

1. On the previous dynamic domain manager launch the following command to export the security file:

   ```
   dumpsec > <file_name>.txt
   ```

2. Copy the `<file_name>.txt` file to the new dynamic domain manager.
3. On the new dynamic domain manager, launch the following command to compile and install the security file:

   ```
   makesec <file_name>.txt
   ```

For more information about the dumpsec and makesec commands, see Updating the security file *(on page)*.

You can now proceed to Installing the new master domain manager configured as a backup *(on page 190)*.

Installing the new master domain manager configured as a backup

You install a master domain manager at the latest product version level configured as the new backup master domain manager by running the serverinst script. The installation process is able to detect the presence of an existing master domain manager and automatically configures this one as the backup master domain manager. The new backup master domain manager is configured to point to the existing database instance.

The IBM® Workload Scheduler administrator installs the master domain manager as the backup. The following information is required:

Table 14. Required information

Required information for performing the installation

Command parameter	Information type	Provided in..
Database information		
--rdbmstype	database type	Upgrading the database for the server components *(on page 239)*
--dbhostname	database hostname	
--dbport	database port	
--dbname	database name	
--dbuser	database user name	
--dbpassword	database password	
IBM® Workload Scheduler information		
--wauser	IBM® Workload Scheduler administrative user name	Creating the IBM Workload Scheduler administrative user *(on page 239)*
--wapassword	IBM® Workload Scheduler administrative user password	
WebSphere Application Server Liberty Base information		
--wlpdir	WebSphere Application Server Liberty Base installation directory	Installing WebSphere Application Server Liberty *(on page 225)*
IBM® Workload Scheduler installation directory		
--inst_dir	installation directory	Current procedure
Security information		
--sslkeysfolder	location of converted certificates	Converting default certificates *(on page 223)*
--sslpassword	password of converted certificates	Converting default certificates *(on page 223)*

You can run the **serverinst** command specifying a typical set of parameters. In this case, default values are used for all remaining parameters. For more information about all **serverinst** parameters and default values, see Server components installation - serverinst script *(on page 310)*.

A properties file named `serverinst.properties` is available if you do not want to type parameters in the command line. This is expecially useful if you need to specify many parameters or if you want to reuse the file for several installations. The file is located in *`image_location`*`/TWS/`*`interp_name`*.

You can specify values in the properties file, type them in the command line, or use both methods. If a parameter is specified both in the properties file and in the command line, the command line value takes precedence.

If you need to modify any of the default values, edit the `serverinst.properties` file, but do not modify the `serverinst.template` file located in the same path.

To install the master domain manager as a backup, perform the following steps:

1. Log in to the workstation where you plan to install the master domain manager.
2. Download the installation images from IBM Fix Central.

3. Start the installation specifying a typical set of parameters. In this case, default values are used for all remaining parameters:

   On Windows operating systems

   ```
   cscript serverinst.vbs --acceptlicense yes --rdbmstype <db_type>
   --dbhostname <db_hostname> --dbport <db_port> --dbname <db_name>
   --dbuser <db_user> --dbpassword <db_password> --wauser <wa_user>
   --wapassword <wa_password> --wlpdir <Liberty_installation_dir>\wlp
   --sslkeysfolder <certificate_files_path> --sslpassword <keystore_truststore_password>
   --inst_dir <installation_dir>
   ```

   On UNIX operating systems

   ```
   ./serverinst.sh --acceptlicense yes --rdbmstype <db_type>
   --dbhostname <db_hostname> --dbport <db_port> --dbname <db_name>
   --dbuser <db_user> --dbpassword <db_password> --wauser <wa_user>
   --wapassword <wa_password> --wlpdir <Liberty_installation_dir>/wlp
   --sslkeysfolder <certificate_files_path> --sslpassword <keystore_truststore_password>
   --inst_dir <installation_dir>
   ```

   where

   --acceptlicense

   Specify **yes** to accept the product license.

   --rdbmstype|-r *rdbms_type*

   The database type. Supported databases are:
   - **DB2**
   - **ORACLE** This value applies to Oracle and Amazon RDS for Oracle
   - **MSSQL** This value applies to MSSQL and MSSQL cloud-based databases.
   - **POSTGRESQL**

   This parameter is required and has no default value.

   --dbhostname *db_hostname*

   The host name or IP address of database server.

   --dbport *db_port*

   The port of the database server.

   --dbname *db_name*

   The name of the IBM® Workload Scheduler database.

   --dbuser *db_user*

   The database user that has been granted access to the IBM® Workload Scheduler tables on the database server.

   --dbpassword *db_password*

   The password for the user that has been granted access to the IBM® Workload Scheduler tables on the database server. Special characters are not supported.

   --wauser *user_name*

   The user for which you are installing IBM Workload Scheduler.

   --wapassword *wauser_password*

   The password of the user for which you are installing IBM Workload Scheduler.

   On Windows operating systems

   Supported characters for the password are alphanumeric, dash (-), underscore (_) characters, and ()|?*~+.@!^

   On UNIX operating systems

   Supported characters for the password are any alphanumeric, dash (-), underscore (_) characters, and ()|?=*~+.

   --wlpdir

The path where WebSphere Application Server Liberty Base is installed.

--sslkeysfolder ***keystore_truststore_folder***

The name and path of the folder containing certificates in PEM format. The installation program automatically processes the keystore and truststore files using the password you specify with the **--sslpassword** parameter. The folder must contain the following files:

- **ca.crt**

  The Certificate Authority (CA) public certificate. Note that if certificates being installed are part of a chain consisting of 3 or more certificates (one Root CA, followed by one or more Intermediate CAs, followed by the end user certificate), then this file must contain the Root CA certificate only. Any Intermediate CA certificates must be stored in the `additionalCAs` subfolder, which therefore becomes a mandatory subfolder. Each Intermediate CA must be stored in the `additionalCAs` subfolder in its own file.

  Note: From V10.2.3, if certificates being installed are part of a chain, the ca.crt can contain also the intermediate CAs. In this case, it must begin with one or more intermediate CA certificates and end with the Root ca.

- **tls.key**

  The private key of the end user certificate for the instance to be installed.

- **tls.crt**

  The public part of the previous key, that is the end user certificate.

For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).

You can optionally create a subfolder to contain one or more `*.crt` files to be added to the server truststore as trusted CA, whose name must be `additionalCAs`. This can be used for example to add to the list of trusted CAs the certificate of the LDAP server or DB2 server. Additionally, you can store here any intermediate CA certificate to be added to the truststore. The subfolder must be named **additionalCAs**. Note that if the end user certificate being installed in the instance is part of a chain consisting of 3 or more certificates (one Root CA, followed by one or more Intermediate CAs, followed by the end user certificate), then the Intermediate CAs certificates must be stored in the `additionalCAs` subfolder, which therefore becomes a mandatory subfolder. Each Intermediate CA must be stored in the `additionalCAs` subfolder in its own file.

For further information about how to generate custom certificates, see Managing certificates using Certman *(on page)*.

--sslpassword ***ssl_password***

The password for the custom certificates and the path to the folder containing certificates in PEM format with the **sslkeysfolder** parameter.

For more information, see sslkeysfolder *(on page 316)*.

You can optionally encrypt the password using the secure script. For more information, see Optional password encryption - secure script *(on page 300)*.

--inst_dir ***installation_dir***

The directory of the IBM Workload Scheduler installation.

Note: The values for the following parameters must match the values you provided when creating and populating the database:

- **--rdbmstype**
- **--dbhostname**

- **--dbport**
- **--dbname**
- **--dbuser**
- **--dbpassword**

Note: Before starting the deployment of a new master domain manager or backup master domain manager on an already used database, be sure that no failed plan creation/extension has been performed. If a failed plan creation or extension has been performed, resolve the failure before attempting the new deployment or unlock the database by running the planman unlock db command.

4. If you are installing a backup master domain manager, it is crucial to use the same encryption keys as those on the master domain manager, to ensure it can correctly decrypt encrypted files, such as the Symphony file. To achieve this, perform the following steps:
   a. Backup the files located in the *TWA_DATA_DIR*`\ssl\aes` folder on the backup master domain manager.
   b. Copy the files from the *TWA_DATA_DIR*`\ssl\aes` folder on the master domain manager to the *TWA_DATA_DIR*`\ssl\aes` folder on the backup master domain manager.
5. Run the following command on the back-level master domain manager to add the new backup master domain manager to the plan:

```
JnextPlan -for 0000
```

You have now successfully installed the master domain manager as the backup master domain manager.

You can now proceed to Switching the manager to the upgraded backup *(on page 194)*.

Switching the manager to the upgraded backup

To switch the back-level manager to the upgraded backup, complete the following procedure:

1. Switch to your upgraded backup manager, which now becomes your current active manager, by issuing the following command from either the Dynamic Workload Console or the **command line** of your back-level manager:

   From the Dynamic Workload Console

   In the navigation tree, click **Monitoring and Reporting > Monitor Workload >** select the engine and the object type Workstation, click **Run** and, in the table of results, select backup manager workstation name, click **More Actions**, and select **Become Master Domain Manager**or **Become Dynamic Domain Manager**, as necessary.

   From the command line of the back-level manager

   Issue the following command:

   ```
   conman "switchmgr masterdm;new_mgr_cpu"
   ```

   where *new_mgr_cpu* is the backup manager workstation name.

2. Switch the event processor from the back-level manager to the backup manager, by running the following command from either the Dynamic Workload Console or the **command line** of your back-level manager:

   From the Dynamic Workload Console

   In the navigation tree, click **Monitoring and Reporting > Monitor Workload >** select the engine and the object type Workstation, click run and, in the table of results, select manager workstation name, click **More Actions**, and select **Become Event Processor**.

   From the command line of the back-level manager

   Issue the following command:

```
conman "switcheventprocessor new_mgr_cpu"
```

where *new_mgr_cpu* is the backup manager workstation name.

You have now successfully switched back-level manager to the upgraded backup.

You can now proceed to make this switch permanent, as described in Making the switch permanent *(on page 195)*.

Making the switch permanent

Making the switch permanent (MDM)

In the procedure Switching the manager to the upgraded backup *(on page 189)*, you switched your manager promoting your new version backup manager to the role of manager.

To make this configuration fully operational and persistent through **JnextPlan**, complete the following procedure on the new manager, referred to as *new_mgr_cpu*:

1. Edit the *localopts* file and modify the following entry as shown:

   ```
   DEFAULTWS=new_mgr_cpu
   ```

   where *new_mgr_cpu* is the workstation name of the new manager. For more information about `localopts` file, see Setting local options *(on page)*.
2. Change the workstation definition of the back-level master by running:

   ```
   composer modify cpu=old_mgr_cpu
   ```

   and in the definition substitute *type=manager* with *type=fta*
3. Change the workstation definition of the new manager by running:

   ```
   composer modify cpu=new_mgr_cpu
   ```

   and in the definition substitute *type=fta* with *type=manager*.
4. Ensure that the **optman** `cf` option is set to *all*.
5. Rebuild the plan to activate the changes to the database:

   ```
   JnextPlan -for 0000
   ```

6. Restore the previous setting of the **optman** `cf` option, if necessary.
7. Edit the */TWA_DATA_DIR/*`mozart/globalopts` file and modify the **master=*old_mgr_cpu*** entry as shown:

   ```
   master=new_mgr_cpu
   ```

   where *new_mgr_cpu* is the workstation name of the new master. For more information about optman, see Setting global options *(on page)*.

   In this way the reports reptr-pre and reptr-post can run when you run **JnextPlan**.

You have now successfully made the switch permanent.

You can now proceed to Customizing and submitting the optional FINAL job stream *(on page 195)*.

Customizing and submitting the optional FINAL job stream

The upgrade process writes the latest FINAL and FINALPOSTREPORTS definitions for the current release in the following file: `<TWA_HOME>/TWS/config/Sfinal`, where `<TWA_HOME>` is the IBM Workload Scheduler installation directory. To use these latest definitions, you must merge the functions of your current FINAL and FINALPOSTREPORTS job streams with the syntax of your new FINAL and FINALPOSTREPORTS job streams.

Important: The definitions of the FINAL and FINALPOSTREPORTS job streams in `<TWA_HOME>/TWS/config/Sfinal` are defined on an extended agent that might not be defined in the new environment. If you are planning to use the old definitions to replace the new ones using the composer command `composer replace`, you must either change the workstation on which the jobs are defined to an existing one, or you must create a new extended agent where the jobs inside the *Sfinal* are defined.

Complete the following procedure:

1. Depending on your situation, edit your current final job streams and customize the new final job streams as follows:

   If you had customized job streams called FINAL and FINALPOSTREPORTS in your database:

   a. Extract the definitions from the current FINAL and FINALPOSTREPORTS job streams file by using composer.
   b. Use a text editor to edit your customized FINAL and FINALPOSTREPORTS job streams.
   c. Merge the job streams with file `<TWA_HOME>/TWS/config/Sfinal` so that the new FINAL and FINALPOSTREPORTS job streams have the same customization as your customized final job streams plus the new required attributes provided by the new FINAL and FINALPOSTREPORTS job streams.
   d. Save your new FINAL and FINALPOSTREPORTS job streams by using composer.

   If you had customized final job streams called something other than FINAL and FINALPOSTREPORTS in your database:

   a. Extract the definitions from your customized final job stream files by using composer.
   b. Use a text editor to edit your customized final job stream files.
   c. Merge the job streams with file `<TWA_HOME>/TWS/config/Sfinal` so that the new FINAL and FINALPOSTREPORTS job streams have the same customization as your customized final job streams plus the new required attributes provided by the new FINAL and FINALPOSTREPORTS job streams.
   d. Save these new final job streams so that they have the same names as your current customized final job streams by running the command composer replace.

   If you had final job streams called something other than FINAL and FINALPOSTREPORTS in your database, but they are not customized:

   a. Make a copy of file `<TWA_HOME>/TWS/config/Sfinal`.
   b. Edit this copy and rename the FINAL and FINALPOSTREPORTS parameters with the actual names.
   c. Run the command composer replace.

   If you had final job streams called FINAL and FINALPOSTREPORTS in your database, but they are not customized:

   Run the command composer replace <TWA_HOME>/TWS/config/Sfinal.

   If you had final job streams called FINAL and FINALPOSTREPORTS but they are in DRAFT in your database:

   Run the command composer replace and, after the upgrade, change these job streams into the DRAFT status again.

2. After you customized the new final job streams, you must delete your current final job stream instances (**conman** cancel sched command) and submit the new final job stream instances (**conman** sbs sched command).

During the upgrade, JnextPlan is overwritten even if you customized it. The existing JnextPlan is backed up and renamed to:

On Windows™ operating systems:

JnextPlan.cmd.bk

On UNIX™ and Linux™ operating systems:

JnextPlan.bk

You have now correctly customized and submitted the optional FINAL job stream.

You can now proceed to Installing a new backup dynamic domain manager *(on page 197)*.

Installing a new backup dynamic domain manager

Procedure for installing a dynamic domain manager configured as a backup.

Install a new dynamic domain manager at the latest product version level configured as the new backup dynamic domain manager by running the serverinst script.

The procedure to install the dynamic domain manager and backup dynamic domain manager is exactly the same, with the difference that it is performed on two different workstations and that each installation points to its local WebSphere Application Server Liberty Base installation. IBM® Workload Scheduler determines whether or not a dynamic domain manager is already present in the environment and proceeds to install a dynamic domain manager or backup dynamic domain manager accordingly.

The IBM® Workload Scheduler administrator installs the dynamic domain manager as the backup. The following information is required:

Table 15. Required information

Required information for performing the installation

Command parameter	Information type	Provided in...
Database information		
--rdbmstype	database type	Upgrading the database for the server components *(on page 239)*
--dbhostname	database hostname	
--dbport	database port	
--dbname	database name	
--dbuser	database user name	
--dbpassword	database password	
IBM® Workload Scheduler information		
--wauser	IBM® Workload Scheduler administrative user name	Creating the IBM Workload Scheduler administrative user *(on page 239)*
--wapassword	IBM® Workload Scheduler administrative user password	
WebSphere Application Server Liberty Base information		
--wlpdir	WebSphere Application Server Liberty Base installation directory	Installing WebSphere Application Server Liberty *(on page 225)*
Security information		
--sslkeysfolder	location of converted certificates	Converting default certificates *(on page 223)*

Table 15. Required information

Required information for performing the installation

(continued)

--sslpassword	password of converted certificates	Converting default certificates *(on page 223)*

Before starting the installation, ensure the following steps have been completed:

1. Converting default certificates *(on page 178)*. Because you are installing a dynamic domain manager, also copy locally the `jwt.crt` file created in the conversion procedure.
2. Upgrading WebSphere Application Server Liberty *(on page 179)* on the workstation where you plan to install the dynamic domain manager and on the workstation where you plan to install the backup dynamic domain manager.
3. Encrypting passwords (optional) *(on page 181)*
4. Upgrading the Dynamic Workload Console and its database *(on page 182)*
5. Creating the IBM Workload Scheduler administrative user *(on page 184)* on the workstations which will host the components at 10.2.3 level.
6. Upgrading the database for the server components *(on page 184)*
7. Customizing and submitting the optional FINAL job stream *(on page 195)*

You can run the **serverinst** command specifying a typical set of parameters. In this case, default values are used for all remaining parameters. For more information about all **serverinst** parameters and default values, see Server components installation - serverinst script *(on page 310)*.

A properties file named `serverinst.properties` is available if you do not want to type parameters in the command line. This is expecially useful if you need to specify many parameters or if you want to reuse the file for several installations. The file is located in *`image_location`*`/TWS/`*`interp_name`*.

You can specify values in the properties file, type them in the command line, or use both methods. If a parameter is specified both in the properties file and in the command line, the command line value takes precedence.

If you need to modify any of the default values, edit the `serverinst.properties` file, but do not modify the `serverinst.template` file located in the same path.

To install the dynamic domain manager as a backup, perform the following steps:

1. Log in to the workstation where you plan to install.
2. Browse to the folder where the `serverinst` command is located:

   On Windows operating systems

   `image_location``\TWS\`*`interp_name`*

   On UNIX operating systems

   `image_location``/TWS/`*`interp_name`*

3. Start the installation specifying a typical set of parameters. In this case, default values are used for all remaining parameters:

   On Windows operating systems

   ```
   cscript serverinst.vbs --acceptlicense yes --rdbmstype db_type
          --dbhostname db_hostname --dbport db_port --dbname db_name
          --dbuser db_user --dbpassword db_password --wauser wa_user
          --wapassword wa_password --componenttype DDM --domain domain_name
          --master mdm_name --mdmbrokerhostname mdm_broker_host_name
          --sslkeysfolder certificate_files_path --sslpassword keystore_truststore_password
          --mdmhttpsport mdm_https_host_name --wlpdir Liberty_installation_dir\wlp
   ```

   On UNIX operating systems

   ```
   ./serverinst.sh --acceptlicense yes --rdbmstype db_type
          --dbhostname db_hostname --dbport db_port --dbname db_name
   ```

```
        --dbuser db_user --dbpassword db_password --wauser wa_user
        --wapassword wa_password --componenttype DDM --domain domain_name
        --master mdm_name --mdmbrokerhostname mdm_broker_host_name
        --sslkeysfolder certificate_files_path --sslpassword keystore_truststore_password
        --mdmhttpsport mdm_https_host_name --wlpdir Liberty_installation_dir/wlp
```

4. Distribute the Symphony file to the new dynamic domain manager configured as backup:
    a. Ensure that the **optman cf** option is set to *all*.
    b. To distribute the Symphony file to the new dynamic domain manager configured as backup, run JnextPlan -for 0000 or wait until the end of the production plan.
    c. Restore the previous setting of the **optman cf** option, if you previously modified the value.

You have now successfully installed the backup dynamic domain manager at the new product version level.

For more information about all **serverinst** parameters and default values, see Server components installation - serverinst script *(on page 310)*.

What to do next

You can now proceed to Cleaning up your environment *(on page 199)*.

Cleaning up your environment

A few final steps towards a clean, efficient environment.

After performing the steps in the procedure, you might want to clean up the environment by performing the following steps:

1. On the new master at version 10.2.3, modify the new master workstation definitions, both fault-tolerant agent and broker, by setting the **SECUREADDR** and **SECURITYLEVEL** parameters.
2. Enable the full SSL global options, as follows:

    ```
    optman chg sf=yes
    ```

3. Run `JnextPlan -for 0000` to make the changes effective.
4. When you installed the new master domain manager as a backup of the master at version 9.5, you installed it without SSL enabled, to allow communication with the back-level environment. You have now to enable SSL again, by switching the value of the **eventProcessorEIFPort (ee)** with the value of the **eventProcessorEIFSSLPort (ef)** global options, and vice versa, as follows

    ```
    optman chg ee=<value_of_ef_option>
    ```

    ```
    optman chg ef=<value_of_ee_option>
    ```

5. If necessary, move all scheduling objects to the new master domain manager and fault-tolerant agent, as follows

    ```
    composer rename <scheduling_object> <old_master_FTA>#@ <new_master_FTA>#@
    ```

6. Edit the job definitions to modify the current STREAMLOGON user with the user of the new master domain manager.
7. On the new master domain manager create a backup of the `BrokerWorkstation.properties` file.
8. Copy the `BrokerWorkstation.properties` file from the previous master domain manager and replace the `BrokerWorkstation.properties` file on the new master domain manager, adjusting every key to the values of the new master domain manager.
9. Modify the **TCPADDR** and **SECUREADDR** parameters in the broker workstation definition by setting the broker port of the new master domain manager.
10. Modify the **eventProcessorEIFSSLPort** global option to the port of the new master domain manager.
11. Stop and restart WebSphere Application Server Liberty.
12. Run `JnextPlan -for 0000` to make the changes effective.

You can now optionally dismiss all back-level components, then proceed to Upgrading agents and domain managers *(on page 200)*.

Upgrading agents and domain managers

There are several methods you can choose from to upgrade your agents and domain managers.

The agent upgrade can be performed with minimal impact to scheduling activities. The agents are stopped for the shortest time necessary to perform the maintenance. Any active agent command-line interfaces and processes, such as `conman, composer, netman, mailman`, and `batchman`, to name a few, continue running. Any jobs already running when the upgrade process begins, continue to run as planned, however, no new jobs begin execution during this time. Once the upgrade is complete, the agent is restarted and quickly reconnects with its jobs. Any jobs that were actively running before the upgrade that have not yet completed, continue to run, and any jobs that successfully finished running during the upgrade procedure report a successful job status. An automatic backup and restore feature is in place in case of failure.

If your agents or domain managers are at version 9.4 or 9.5, you can upgrade directly to version 10.2.3.

If you choose to upgrade your environment top-down, then the agents get upgraded progressively after you have upgraded the master domain manager and its backup. This means that new features and enhancements are not available on all of your agents at the same time. If, instead, you choose to upgrade your environment bottom-up, then the agents are upgraded first, and new features and enhancements become available after themaster domain manager and its backup have been upgraded.

Important: After upgrading your fault-tolerant agents, it might be necessary to manually update the security file on the fault-tolerant agents in your environment to add access to folders for all of the scheduling objects that can be defined or moved into folders. These updates are especially important if you plan to use the command line on the fault-tolerant agents to perform operations on the objects in folders. See Updating the security file *(on page)* for more information.

You can choose to upgrade your agents using any of the following methods:

twsinst script

A single line command that checks if processes or a command line is running before it starts. It saves disk space and RAM because it is not Java-based. See Upgrade procedure *(on page 201)* and Upgrading agents on IBM i systems *(on page 205)*

Centralized agent update

Upgrade or update multiple fault-tolerant agent and dynamic agent instances at the same time. Download the fix pack installation package, or the eImage upgrade package to the master domain manager and then either run the installation on multiple agent instances or schedule the installation by creating and submitting a job to run. This upgrade method is not supported on z-centric agent instances. See Centralized agent update *(on page 208)*.

HCL BigFix

Upgrade IBM® Workload Scheduler agents using HCL BigFix analyses and fixlets. You can choose to schedule the upgrade or you can run it immediately. See Upgrading agents using HCL BigFix *(on page 216)*.

For a list of supported operating systems and requirements, see the System Requirements Document at IBM Workload Scheduler Detailed System Requirements.

When the upgrade procedure has completed successfully, the backup instance is deleted.

Note: The `localopts` file is not modified during the agent upgrade process. The file generated by the upgrade process is saved to the `/config` directory to maintain you custom values, if any. You can then merge the two files with your customized values and save the resulting file in the following path:

On Windows operating systems

<TWA_home>\TWS

On UNIX operating systems

<TWA_DATA_DIR>

When upgrading dynamic agents featuring both a local and a remote gateway, ensure you either upgrade the agent first and then the gateway or upgrade both at the same time.

After completing the upgrade, you can optionally install a new backup master domain manager at version 10.2.3 to ensure failover capabilities.

Upgrade procedure

1. Verify that the user running the process has the following authorization requirements:

   Windows™ operating systems

   If you set the Windows User Account Control (UAC), your login account must be a member of the Windows™ **Administrators** group or domain administrators with the right, **Act as Part of the Operating System**.

   If you set the Windows User Account Control (UAC) on the workstation, you must run the installation as **administrator**.

   UNIX™ and Linux™ operating systems

   If the component was installed with root privileges, **root** access is required. If you performed a **no-root installation**, specify the same user used for installing the component.
2. Download the installation images from IBM Fix Central.
3. Ensure that you have enough temporary space before starting the installation process.

To upgrade agents, from the directory that contains the IBM Workload Scheduler agent eImage, run the **twsinst** script using the synopsis described below.

twsinst for Windows™ is a Visual Basic Script (VBS) that you can run in CScript and WScript mode, for example:

```
cscript twsinst.vbs -update -uname user_name -acceptlicense yes -enablefips false
```

On UNIX operating systems, the syntax is as follows:

```
./twsinst -update -uname user_name -acceptlicense yes -enablefips false
```

Synopsis:

Windows™ operating systems

UNIX™ and Linux™ operating systems

Show command usage and version

```
./twsinst -u | -v
```

Upgrade an instance

```
./twsinst -update [-uname user_name]
 -acceptlicense yes|no
 [-addjruntime true]
 [-create_link]
 [-inst_dir install_dir [-recovInstReg true]]
 [-lang lang-id]
 [-reset_perm]
 [-patch]
 [-skipbackup]
 [-skipcheckprereq]
 [-skip_usercheck]
 [-wait minutes]
 [-work_dir working_dir]
 [--enablefips true | false]
```

-acceptlicense *yes|no*

Specify whether or not to accept the License Agreement.

-addjruntime *true*

Adds the Java™ run time to run job types with advanced options to the agent. The run time environment is used to run application job plug-ins on the agent and to enable the capability to run remotely, from the agent, the dynamic workload broker resource command on the server.

This option is applicable to both fault-tolerant agents and dynamic agents.

By default, if the Java run time was already installed on the agent, it is upgraded.

If the Java run time was not installed on the agent, it is not installed during the upgrade, unless you specify `-addjruntime true`.

If you decided not to install the Java™ run time when you upgrade, you can add this feature later, as described in Adding a feature *(on page 152)*.

-create_link

UNIX™ operating systems only. Create the **symlink** between `/usr/bin/at` and *`install_dir`*`/TWS/bin/at`. For more information, see Table 2: Symbolic link options *(on page 22)*.

-enablefips

Specify whether you want to enable FIPS. In the current product version, you can only specify `false` because FIPS is not supported. In a fresh installation, the default is `false`. In upgrade, there is no default value, so you have to set it explicitly and be aware that FIPS is being disabled when you upgrade. This parameter is optional. If you are upgrading from an environment where FIPS is supported, see Q: My environment is FIPS compliant. What happens if I upgrade to version 10.2.3? *(on page 270)*.

-inst_dir *install_dir*

The directory where you installed IBM Workload Scheduler. When upgrading, the directory **inst_dir** is used whether:

- The upgrade process cannot retrieve the product install location from the registries.
- You need to create the IBM Workload Scheduler registries again before upgrading. See Re-creating registry files using twsinst *(on page 263)* for details.

If you do not provide the **inst_dir** directory and IBM Workload Scheduler cannot retrieve it from the installation registries, the product is installed in the user home directory.

On Windows™ operating systems:

If you specify a path that contains blanks, enclose it in double quotation marks. If not specified, the path is set to `%ProgramFiles%\IBM\TWA`.

On UNIX™ and Linux™ operating systems:

The path cannot contain blanks. If not specified, the path is set to the *user_name* home directory.

-lang

The language in which the `twsinst` messages are displayed. If not specified, the system LANG is used. If the related catalog is missing, the default C language catalog is used.

Note: The **-lang** option does not relate to the supported language packs. By default, all supported language packs are installed when you install using the twsinst script.

-password

Windows system only. The password of the user for which you are installing IBM Workload Scheduler. The password is not required for the upgrade procedure.

-recovInstReg *true*

To re-create the registry files. Specify if you tried to upgrade a stand-alone, fault-tolerant agent (an agent that is not shared with other components or does not have the connector feature) and you received an error message that states that an instance of IBM Workload Scheduler cannot be found. This error can be caused by a corrupt registry file. See Upgrading when there are corrupt registry files *(on page 263)*. If you specify this parameter you must set **-inst_dir** option.

-reset_perm

UNIX™ systems only. Reset the permissions of the libatrc library.

-skipcheckprereq

If you specify this parameter, IBM Workload Scheduler does not scan system prerequisites before installing the agent. For more information on the prerequisite check, see Scanning system prerequisites for IBM Workload Scheduler *(on page 33)*.

- patch

Specifies that a patch must be installed. When you specify this option, only the files present in the patch package are replaced in the installed product and all other product files remain unchanged.

-skipbackup

If you specify this parameter the upgrade process does not create a backup of the instance you are upgrading. If the agent upgrade fails, the agent cannot be restored. If you do not specify this parameter, the upgrade process creates a backup of the agent instance in the path `work_dir>/backup`. The *work_dir>* is a temporary directory used by the upgrade process. It can be defined by passing the parameter -work_dir to the twsinst script. If you do not define the work_dir then by default it is set to `/tmp/TWA_${INST_USER}/tws94`, where `tmp` is the temporary directory of the operating system and ${*INST_USER*} is the user performing the upgrade. For example, `/tmp/TWA_jsmith/tws94/backup`.

-skip_usercheck

Enable this option if the authentication process within your organization is not standard, thereby disabling the default authentication option. On UNIX™ and Linux™ operating systems if you specify this parameter, the program skips the check of the user in the `/etc/passwd` file or the check you perform using the su command. On Windows™ operating systems if you specify this parameter, the program does not create the user you specified in the -uname *username* parameter. If you specify this parameter you must create the user manually before running the script.

-uname *username*

The name of the user for which IBM Workload Scheduler is being updated. The software is updated in this user's home directory. This user name is not to be confused with the user performing the upgrade.

-update

Upgrades an existing agent that was installed using the **twsinst** script.

-wait ***minutes***

The number of minutes that the product waits for jobs that are running to complete before starting the upgrade. If the jobs do not complete during this interval the upgrade does not proceed and an error message is displayed. Valid values are integers or **-1** for the product to wait indefinitely. The default is **60**.

-work_dir ***working_dir***

The temporary directory used for the IBM Workload Scheduler upgrade process files deployment.

On Windows™ operating systems:

If you specify a path that contains blanks, enclose it in double quotation marks. If you do not manually specify a path, the path is set to `%temp%\TWA\tws<version_number>`, where `%temp%` is the temporary directory of the operating system.

On UNIX™ and Linux™ operating systems:

The path cannot contain blanks. If you do not manually specify a path, the path is set to `/tmp/TWA/tws<version_number>`.

When the agent upgrade completes, the agent is restarted and quickly reconnects with its jobs. Any jobs that were actively running before the upgrade that have not yet completed, continue to run, and any jobs that successfully finished running during the upgrade procedure report a successful job status. An automatic backup and restore feature is in place in case of failure.

Examples

This section contains examples of **twsinst** scripts that you can use to upgrade an agent.

To upgrade an agent installed in the user home directory that does not have the dynamic scheduling capabilities and the Java™ run time to run job types with advanced options:

```
./twsinst -update -uname twsuser -acceptlicense yes
```

To upgrade an agent installed in the path `/opt/IBM/TWA` on UNIX operating systems and in the path `C:\Program Files\IBM\TWA` on Windows operating systems, and give it dynamic scheduling capabilities, but not the Java™ run time to run job types with advanced options:

On Windows™ operating systems:

```
cscript twsinst -update -uname TWS_user -password password
-acceptlicense yes
-tdwbhostname mybroker.mycompany.com -tdwbport 31116
-inst_dir "c:\Program Files\IBM\TWA"
```

On UNIX™ and Linux™ operating systems:

```
./twsinst -update -uname twsuser
-acceptlicense yes
-tdwbhostname mybroker.mycompany.com
-tdwbport 31116 -inst_dir /opt/IBM/TWA
```

To upgrade an agent and give it both dynamic scheduling capabilities and the Java™ run time to run job types with advanced options. The run time environment is used to run application job plug-ins on the agent and to enable the capability to remotely run, from the agent, the dynamic workload broker resource command on the server:

On Windows™ operating systems:

```
cscript twsinst -update -uname TWS_user -password password
-acceptlicense yes
-tdwbhostname mybroker.mycompany.com -tdwbport 31116 -addjruntime true
-inst_dir "c:\Program Files\IBM\TWA"
```

On UNIX™ and Linux™ operating systems:

```
./twsinst -update -uname twsuser -acceptlicense yes
-tdwbhostname mybroker.mycompany.com
-tdwbport 31116 -addjruntime true
```

Upgrading agents on IBM i systems

How to upgrade agents on IBM i systems.

You can upgrade the agent on an IBM i system by using the `twsinst` installation script.

To upgrade an IBM Workload Scheduler agent, perform the following steps:

1. Sign on as the user who performed the installation, either **QSECOFR** or an existing user with ALLOBJ authority. If you installed with a user different from **QSECOFR**, use the same user who performed the installation and specify the **allObjAuth** parameter to indicate that the user has the ALLOBJ authority. For more information about this parameter, see Agent installation parameters on IBM i systems *(on page 112)*. You can find the name of the profile used to perform the installation in the `instUser` located in the *`agent_data_dir`*`/installation/instInfo`.
2. Download the installation images from IBM Fix Central.
3. If you downloaded the eImages, to extract the package, use the *PASE* shell or the *AIXterm* command.

   Using *PASE* shell:

   a. Open the *PASE* shell.
   b. Run the command "CALL QP2TERM".
   c. Locate the folder where you downloaded the eImages and run the command:

   ```
   "tar xvf TWS1023_IBM_I.tar"
   ```

   d. Exit from the *PASE* shell.

   Using *AIXterm* command:

   a. Start the *Xserver* on your desktop.
   b. On the iSeries machine, open a *QSH shell* and export the display.
   c. In *QSH shell*, go to the directory */QopenSys* and run the command "aixterm -sb".
   d. A pop-up window is displayed on your desktop. By Using this pop-up window, extract the file *TWS1023_IBM_I.tar*.

4. Open a *QSH shell* and run the **twsinst** script.

   The installation procedure replaces the library to the user profile library list of the dynamic agent user profile and sets this job description as the job description of the dynamic agent user profile. The upgrade process replaces the new version of the agent in the directory where the old agent is installed.

If the operation fails to understand the cause of the error, see Analyzing return codes for agent installation, upgrade, restore, and uninstallation *(on page 280)*.

Command usage and version

Show command usage and version

```
twsinst -u | -v
```

Upgrade an instance

```
./twsinst -update -uname user_name
 -acceptlicense yes|no
 [-addjruntime true]
 [-allObjAuth]
 [-create_link]
 [-hostname host_name]
 [-inst_dir install_dir]
 [-jmport port_number]
 [-jmportssl boolean]
 [-lang lang-id]
 [-reset_perm]
 [-recovInstReg true]
 [-skip_usercheck]
 -tdwbhostname host_name
 -tdwbport port_number
 [-wait minutes]
 [-work_dir working_dir]
```

For a description of the installation parameters and options that are related to agent on this operating system, see Agent upgrade parameters on IBM i systems *(on page 206)*.

Agent upgrade parameters on IBM i systems

The parameters set when using the **twsinst** script to upgrade a dynamic agent on IBM i systems.

-acceptlicense *yes|no*

Specify whether to accept the License Agreement.

-addjruntime *true*

Adds the Java™ run time to run job types with advanced options to the agent. The run time environment is used to run application job plug-ins on the agent and to enable the capability to run remotely, from the agent, the dynamic workload broker resource command on the server.

By default, if the Java run time was already installed on the agent, it will be upgraded to the new version.

If the Java run time was not installed on the agent, it will not be installed during the upgrade, unless you specify `-addjruntime true`.

If you decide not to install Java™ run time when upgrading, you can still add this feature at a later time as it is described in Adding a feature *(on page 152)*.

-allObjAuth

If you are installing, upgrading, or uninstalling with a user different from the default **QSECOFR** user, this parameter specifies that the user has the required ALLOBJ authority. Ensure the user is existing and has ALLOBJ authority because the product does not verify that the correct authority is assigned. The same user must be specified when installing, upgrading or uninstalling the agent. If you are using the **QSECOFR** user, this parameter does not apply.

-create_link

Create the **symlink** between `/usr/bin/at` and `<install_dir>/TWS/bin/at`. See Table 2: Symbolic link options *(on page 22)* for more information.

-displayname

The name to assign to the agent. The default is the host name of this computer.

-inst_dir *installation_dir*

The directory of the IBM Workload Scheduler installation.

Note: The path cannot contain blanks. If you do not manually specify a path, the path is set to the default home directory, that is, the *user_ home\user_name* directory.

-jmport *port_number*

The JobManager port number used by the dynamic workload broker to connect to the IBM Workload Scheduler dynamic agent. The default value is **31114**. The valid range is from 1 to 65535.

-jmportssl *true|false*

The JobManager port used by the dynamic workload broker to connect to the IBM Workload Scheduler dynamic agent. This number is registered in the `ita.ini` file located in the `ITA/cpa/ita` directory.

For communication using SSL or HTTPS

Set **jmportssl = true**. To communicate with the dynamic workload broker, it is recommended that you set the value to **true**. If the value is set to *true*, the port specified in **jmport** communicates in HTTPS.

For communication without using SSL, or through HTTP

Set **jmportssl = false**. If the value is set to *false*, the port specified in **jmport** communicates in HTTP.

-lang ***lang_id***

The language in which the `twsinst` messages are displayed. If not specified, the system LANG is used. If the related catalog is missing, the default C language catalog is used.

Note: This is the language in which the installation log is recorded, and not the language of the installed engine instance. The twsinst script installs all languages by default.

-recovInstReg ***true***

To re-create the registry files. Specify it if you have tried to upgrade a stand-alone, fault-tolerant agent (an agent that is not shared with other components or does not have the connector feature) and you received an error message that states that an instance of IBM Workload Scheduler cannot be found, this can be caused by a corrupt registry file. See Upgrading when there are corrupt registry files *(on page 263)*.

-skip_usercheck

Enable this option if the authentication process within your organization is not standard, thereby disabling the default authentication option. If you specify this parameter, you must create the user manually before running the script.

-skipcheckprereq

If you specify this parameter, IBM Workload Scheduler does not scan system prerequisites before upgrading the agent.

For a detailed list of supported operating systems and product prerequisites, see IBM Workload Scheduler Detailed System Requirements.

-tdwbhostname ***host_name***

The dynamic workload broker fully qualified host name. It is used together with the **-tdwbport** *tdwbport_number* parameter. It adds and starts the capabilities to run workload dynamically to IBM Workload Scheduler. This value is registered in the **ResourceAdvisorUrl** property in the `JobManager.ini` file.

-tdwbport ***tdwbport_number***

The dynamic workload broker HTTP or HTTPS port number used to add dynamic scheduling capabilities to your distributed or end-to-end environment. It is used together with the **-tdwbhostname** *host_name* parameter. This number is registered in the ResourceAdvisorUrl property in the `JobManager.ini` file. Specify a nonzero value to add dynamic capability. The valid range is 0 to 65535.

-uname ***user_name***

The name of the user for which IBM Workload Scheduler is being updated. The software is updated in this user's home directory. This user name is not to be confused with the user performing the upgrade.

Note: This user name is not the same as the user performing the installation logged on as **QSECOFR**.

-update

Upgrades an existing agent that was installed using **twsinst**.

-wait ***minutes***

The number of minutes that the product waits for jobs that are running to complete before starting the upgrade. If the jobs do not complete during this interval the upgrade does not proceed and an error message is displayed. Valid values are integers or **-1** for the product to wait indefinitely. The default is **60** minutes.

-work_dir ***working_dir***

The temporary directory used for the IBM Workload Scheduler installation process files deployment. The path cannot contain blanks. If you do not manually specify a path, the path is set to `/tmp/TWA/tws1023`.

Example upgrade of an agent on IBM i systems

The following example shows the syntax used when using the **twsinst** script to upgrade an instance of the agent on IBM i system.

```
./twsinst -update
-uname TWS_user
-allObjAuth
-acceptlicense yes
-nobackup
-work_dir "/tmp/TWA/tws1023"
```

The twsinst script log files on IBM i systems

The twsinst log file name is:

Where: `<TWS_INST_DIR>/twsinst_IBM_i_TWS_user^product_version.log`

TWS_INST_DIR

The IBM Workload Scheduler installation directory. The default installation directory is `/home/TWS_user`.

TWS_user

The name of the user for which IBM Workload Scheduler was installed, that you supplied during the installation process.

product_version

Represents the product version. For example, for version 10.2.3 of the product, the value is 10.2.3.00

Centralized agent update

You can install fix packs or upgrade releases for multiple fault-tolerant agent and dynamic agent instances, by downloading a package on the master domain manager workstation and updating the multiple agent instances by running an action from the Dynamic Workload Console.

You can also schedule the centralized update of multiple agent instances, by using the Dynamic Workload Console or the command line.

The centralized agent update process does not apply to z-centric agents. Also, a distributed master domain manager is required.

During the upgrade or update, the agents are stopped for the shortest time necessary to perform the maintenance. Any active agent command-line interfaces and processes, such as conman, composer, netman, mailman, and batchman, to name a few, continue running. Any jobs already running when the upgrade process begins, continue to run as planned, however, no new jobs begin execution during this time. Once the upgrade is complete, the agent is restarted and quickly reconnects with its jobs. Any jobs that were actively running before the upgrade that have not yet completed, continue to run, and any jobs that successfully finished running during the upgrade procedure report a successful job status.

You can find the full procedure in Centralized agent update by using Dynamic Workload Console *(on page 208)*.

Note:

Avoid installing multiple agents (fault-tolerant agents or dynamic agents) at the same time on the same system, because this could cause the installation to fail.

For the latest information about Centralized agent update, see the Release Notes available at IBM Workload Scheduler Release Notes.

Centralized agent update by using Dynamic Workload Console

You can centrally update multiple fault-tolerant agent and dynamic agent instances with just one single action by using Dynamic Workload Console.

For more information about the `manage` keyword usage, see Object type - cpu *(on page)*. For an example of a master domain manager `Security` file, see Security file on the master domain manager to install fix packs or upgrade fault-tolerant agents and dynamic agents *(on page)*.

Complete the following steps:

1. From the installation package download site, download on the master domain manager workstation the fix pack or upgrade installation package that you want to install on fault-tolerant agent or dynamic agent instances in the following default directory:

   On Windows operating systems:

   `<TWA_home>\TWS\depot\agent`

   On UNIX operating systems:

   `<TWA_home>/TWS/depot/agent`

   where *TWA_home* is the master domain manager installation directory.

   You can change the default directory value performing the following steps:
   - Stop WebSphere Application Server Liberty Base on the master domain manager
   - Modify the `com.ibm.tws.conn.engine.depot` key value in the following property file:

     On Windows operating systems:

     ```
     TWA_home>\usr\servers\engineServer\resources\properties
     \TWSConfig.properties
     ```

     On UNIX operating systems:

     ```
     TWA_home>/usr/servers/engineServer/resources/properties/
     TWSConfig.properties
     ```

   - Start WebSphere Application Server Liberty Base

   Ensure the installation files are readable by the operating system user which owns the Application Server process (java).
2. Log on to Dynamic Workload Console.
3. Create a `Monitor Workstations` task, as described in Creating a task to Monitor Workstations *(on page)*.
4. Run a `Monitor Workstations` task and select one or more dynamic agent or fault-tolerant agent instances that you want to update.
5. Click **More Actions > Update agent**. The Update agent action checks whether the selected agent is a supported workstation type.

   The Update agent action is applicable to the following workstation types only:
   - Dynamic Agent
   - Fault-tolerant agent

   The Update agent action is not applicable to the following workstation types:
   - Master domain manager
   - Backup master domain manager
   - Dynamic domain manager
   - Backup dynamic domain manager
   - Extended agent
   - Standard agent
   - Remote engine
   - Broker
   - Pool
   - Dynamic pool
   - Limited fault-tolerant agent

   The process updates the agent only if the workstation type is supported. Otherwise, either an error message is displayed on the Dynamic Workload Console, or is written in the operator log messages console, depending on the workstation type.

You can schedule the centralized update of multiple agent instances, by using the Dynamic Workload Console or the command line. For a description of the scheduling option, see: Scheduling the centralized agent update *(on page 210)*.

For a description of the **Update agent** action on fault-tolerant agents and dynamic agents, see: Updating fault-tolerant agent and dynamic agent instances *(on page 211)*.

Verify the update agent results by completing one of the following actions in the Dynamic Workload Console:

Check the operator log messages console:

Click **Monitoring and Reporting > Event Monitoring > Monitor Triggered Actions** and check the messages related to the agent workstation update.

The following event rules are triggered:

UPDATESUCCESS

When the workstation is successfully updated

UPDATEFAILURE

When an error occurs

UPDATERUNNING

With the information about the update process status

Check the workstation version changes:

After the next plan update, in the `Monitor Workstations` view of the Dynamic Workload Console, you can check the updated version in the `Version` column of the selected agent. Otherwise, if you do not want to wait for the next plan update to see the updated version, run the command **JnextPlan -for 0000** with the **-noremove** option.

You can also perform a **manual check of the update agent results** by looking at the following log files on the agent system:

On Windows operating systems:

`<TWA_home>\logs\centralized_update.log`

On UNIX operating systems:

`<TWA_home>/logs/centralized_update.log`

Scheduling the centralized agent update

You can schedule the centralized update of multiple agent instances by creating a centralized agent update job, either by using the Dynamic Workload Console or the **composer** command line.

Creating a centralized agent update job by using the Dynamic Workload Console:

1. Log on to the Dynamic Workload Console.
2. Create a `Centralized agent update` job type definition, as described in Managing job definitions *(on page)* .
3. In the properties panel, specify the attributes for the job definition that you are creating. For all the details about available fields and options, see the online help by clicking the "?" in the upper-right corner.
4. In the `Connection` tab, specify the master domain manager workstation where you loaded the fix pack installation package, or the upgrade eImage, that you want to install on fault-tolerant agent or dynamic agent instances.
5. In the `Action` tab, define the list of fault-tolerant agent or dynamic agent instances that you want to update. You can select up to 20 agent instances.
6. Save the job definition in the database.

Creating a centralized agent update job by using the composer command line:

This section describes the required and optional attributes that you need to specify to create a centralized agent update job by using the **composer** command line. For more information, see Job definition *(on page)*:

Table 16. Required and optional attributes for the definition of a centralized agent update job

Attribute	Description and value	Required
hostname	The host name of the master domain manager workstation where you loaded the fix pack installation package, or the upgrade eImage, that you want to install on fault-tolerant agent or dynamic agent instances.	✓
port	The port number of the master domain manager workstation.	✓
protocol	The protocol for connecting to the master domain manager workstation. Supported values are **http** and **https**.	✓
userName	The user to be used for accessing the master domain manager workstation. This attribute is optional, depending on your settings.	
password	The password to be used for accessing the master domain manager workstation. This attribute is optional, depending on the settings on your server.	
NumberOfRetries	The number of times the program tries to connect to the master domain manager workstation. Default value is 0.	
RetryIntervalSeconds	The number of seconds the program waits before retrying the operation. Default value is 30 seconds.	

workstationListValues	The list of agent instances that you want to update.
	Example:
	`<jsdlcentralizedagentupdate:workstationsListValue>`
	`NY053015_AGT (type: Agent, version: 9.5.0.00)`
	`</jsdlcentralizedagentupdate:workstationsListValue>`
	`<jsdlcentralizedagentupdate:workstationsListValue>`
	`NY053009_AGT (type: Agent, version: 9.5.0.00)<`
	`/jsdlcentralizedagentupdate:workstationsListValue>`
	`<jsdlcentralizedagentupdate:workstationsListValue>`
	`NY053016_FTA (type: FTA, version: 9.5.0.00)`
	`</jsdlcentralizedagentupdate:workstationsListValue>`
	You can specify up to 20 agent instances.
		✓

Scheduling a centralized agent update job

You can schedule a centralized agent update job by adding the necessary scheduling arguments to your job, and submitting it. You can submit jobs by using the Dynamic Workload Console or the **conman** command line.

When the job runs, the job forwards to the master domain manager the Update agent request for all the fault-tolerant agent or dynamic agent instances that you selected, and then completes.

Note: The job does not wait for the Update agent request to complete. The completion status of the centralized agent update job refers only to the submission of the Update agent request; the completion status does not refer to the agent update results. To verify the agent update results, see the *Results* section in Centralized agent update by using Dynamic Workload Console *(on page 208)*.

Job properties

When the job completes, you can see the job properties by running:

`conman sj job_name;jobprop`

where*job_name* is the centralized agent update job name.

The following example shows the `Extra Information` section of the output command:

```
EXTRA INFORMATION
The update request has been successfully submitted for the following workstations:
NY053015_AGT|NY053009_AGT|NY053016_FTA
```

Updating fault-tolerant agent and dynamic agent instances

A description of the **Update agent** action on fault-tolerant agents and dynamic agents.

When you run the `update agent` action in the `Monitor Workstations` task from Dynamic Workload Console, or when you schedule a centralized agent update job, IBM Workload Scheduler completes the following steps:

1. The fix pack or upgrade installation package is copied to the master domain manager workstation, and its content is extracted to the following default directory:

   For fault-tolerant agent workstations:

On Windows™ operating systems:

`<TWA_home>\TWS\stdlist\download`

On UNIX™ operating systems:

`<TWA_home>/TWS/stdlist/download`

For dynamic agent workstations:

On Windows™ operating systems:

`<TWA_home>\TWS\stdlist\JM\download`

On UNIX™ operating systems:

`<TWA_home>/TWS/stdlist/JM/download`

Where *TWA_home* is the fault-tolerant agent or dynamic agent installation directory. You can change this default directory by modifying the `DownloadDir` value in the following configuration file:

For fault-tolerant agent workstations:

On Windows™ operating systems:

`<TWA_home>\localopts`

On UNIX™ operating systems:

`<TWA_DATA_DIR>/TWS/localopts`

For dynamic agent workstations:

On Windows™ operating systems:

`<TWA_home>\TWS\ITA\cpa\config\JobManager.ini`

On UNIX™ operating systems:

`<TWA_DATA_DIR>/ITA/cpa/config/JobManager.ini`

Note:

If the path specified in `DownloadDir` does not exist, a warning message is issued and the default download directory is used.

If you are updating both fault-tolerant agent and dynamic agent instances on the same workstation, be sure that you specify different download directories.

2. On the agent workstation, the following script runs automatically:

For fault-tolerant agent workstations:

On Windows™ operating systems:

`<TWA_home>\TWS\stdlist\download\.self\selfupdate.wsf`

On UNIX™ operating systems:

`<TWA_DATA_DIR>/stdlist/download/.self/selfupdate.sh`

For dynamic agent workstations:

On Windows™ operating systems:

`<TWA_home>\TWS\stdlist\JM\download\.self\selfupdate.wsf`

On UNIX™ operating systems:

`<TWA_DATA_DIR>/stdlist/JM/download/.self/selfupdate.sh`

The centralized agent update script, named **selfupdate**, performs a backup of the agent workstation, runs the **twsinst** installation command, and creates the following log file:

On Windows™ operating systems:

`<TWA_home>\TWS\logs\centralized_update.log`

On UNIX™ operating systems:

`<TWA_DATA_DIR>/TWS/logs/centralized_update.log`

Note:

If for any reason the agent update fails, the **selfupdate** script restores the agent to its initial status. The backup files are removed after the agent update completes successfully. The backup files are not removed when the agent restore fails or is successful. For more information about restoring agent instances, see the troubleshooting scenario Manually restore agent instances when the automatic restore fails *(on page 214)*. To modify the backup directory, specify the new directory in the BACKUP_DIR variable in the selfupdate.wsf script.

Troubleshooting scenarios

You can troubleshoot the centralized agent update.

You can troubleshoot the centralized agent update by reading the centralized_update log file.

Prerequisite scan detects missing prerequisites and the centralized agent update fails

You are centrally updating dynamic agents or fault-tolerant agents but the prerequisite scan detects missing prerequisites and the agent installation fails.

Cause and solution

The centralized agent update fails because the prerequisite scan detects missing prerequisites. In this case, analyze the prerequisite scan log file and solve the error, if any. You can then decide to rerun the installation or upgrade without executing the prerequisite scan. To do this, perform the following steps:

1. On the master domain manager workstation, go to the directory where you download the fix pack installation package, or the eImage that you want to install on the agent. The default directory value is:

   On Windows operating systems:

   `<TWA_home>\TWS\depot\agent`

   On UNIX operating systems:

   `<TWA_home>/TWS/depot/agent`

   where *TWA_home* is the master domain manager installation directory.
2. Edit the following script:

   On Windows operating systems:

   ```
   <TWA_home>\TWS\depot\agent\TWS1023_agent_platform_AGENT.zip\.self
   \selfupdate.wsf
   ```

   On UNIX operating systems:

   ```
   <TWA_home>/TWS/depot/agent/TWS1023_agent_platform_AGENT.zip/.self/
   selfupdate.sh
   ```
3. In the selfupdate script, locate the twsinst installation command and add the `-skipcheckprereq` option. If you specify the **-skipcheckprereq** parameter, the twsinst script does not execute the prerequisite scan. For more information about the `-skipcheckprereq` option, see Agent installation parameters - twsinst script *(on page 84)*.

Centralized agent update fails because the temporary backup directory is too small

You are centrally updating dynamic agents or fault-tolerant agents but the backup directory used is too small, and the agent installation fails.

Cause and solution

The centralized agent update fails because the backup directory, by default */tmp*, does not have enough space. You can set a different directory by performing the following steps:

1. On the master domain manager workstation, go to the directory where you downloaded the fix pack installation package, or the eImage that you want to install on the agent. The default directory value is:

   On Windows operating systems:

   `<TWA_home>\TWS\depot\agent`

   On UNIX operating systems:

   `<TWA_home>/TWS/depot/agent`

   where *TWA_home* is the master domain manager installation directory.
2. Edit the following script:

   On Windows operating systems:

   ```
   <TWA_home>\TWS\depot\agent\TWS1023_agent_platform_AGENT.zip\self
   \selfupdate.wsf
   ```

   On UNIX operating systems:

   ```
   <TWA_home>/TWS/depot/agent/TWS1023_agent_platform_AGENT.zip/self/
   selfupdate.sh
   ```
3. In the selfupdate script, locate the BACKUP_DIR variable and replace the value to the directory you want to use as backup.

Note: This directory will be removed at the end of the installation.

Manually restore agent instances when the automatic restore fails

You are upgrading dynamic agents or fault-tolerant agents using either the centralized agent update method or the twsinst script, but the update process fails and starts the automatic restore process. If the automatic restore process fails, you need to manually restore the old agent instances.

Cause and solution

The automatic restore process might fail for several causes, for example, the automatic process does not have the necessary space to perform the operation. If you want to manually restore the old agent instance, complete the following steps:

1. On the workstation where the agent is installed, go to the temporary directory, where the selfupdate script backs up the agent installation directory. The default temporary directory value is:

   Centralized agent update method

   On Windows operating systems:

   `%TEMP%\backupTWS\date`

   On UNIX operating systems:

   `/tmp/backupTWS/date`

   Where *date* is the date of the selfupdate running for your agent instance.

   twsinst script upgrade method

   On Windows operating systems:

   `<working_dir>>\backupTWS\date`

   On UNIX operating systems:

   `<working_dir>>/backupTWS/date`

where *working_dir*> is a temporary directory used by the upgrade process. You define the *working_dir*> passing the **-work_dir** parameter to the twsinst script. If you do not define the *working_dir*> then by default it is set to `/tmp/TWA_${INST_USER}/tws94`, where tmp is the temporary directory of the operating system and ${*INST_USER*} is the user performing the upgrade. For example, on a UNIX operating system: `/tmp/TWA_jsmith/tws94/backup`.

Where *date* is the date of the selfupdate running for your agent instance.

2. Locate the *agent_instance_backup_dir* backup directory for your agent instance.
3. Copy the content of the following directory to the `TWS_agent_inst_dir` agent installation directory:

Centralized agent update method

On Windows operating systems:

`%TEMP%\backupTWS\`*`date`*`\`*`agent_instance_backup_dir`*

On UNIX operating systems:

`/tmp/backupTWS/`*`date/agent_instance_backup_dir`*

twsinst script upgrade method

On Windows operating systems:

`<`*`working_dir`*`>\backupTWS\`*`date`*`\`*`agent_instance_backup_dir`*

On UNIX operating systems:

`<`*`working_dir`*`>/backupTWS/`*`date/agent_instance_backup_dir`*

4. In the `TWS_agent_inst_dir` directory, re-create the `stdlist` directory.
5. Manually delete the following lock file:

Centralized agent update method

On Windows operating systems:

`%TEMP%\twsselfupdate.lock`

On UNIX operating systems:

`/tmp/twsselfupdate.lock`

twsinst script upgrade method

On Windows operating systems:

`<`*`working_dir`*`>\twsselfupdate.lock`

On UNIX operating systems:

`<`*`working_dir`*`>/twsselfupdate.lock`

6. Restart the agent instance.

Centralized agent update does not complete and no operator message is displayed

You are centrally updating dynamic agents and fault-tolerant agents from Dynamic Workload Console. An agent is in running status in the Dynamic Workload Console, but the update process does not complete and no operator message is displayed.

Cause and solution

The agent has been stopped but the Dynamic Workload Console has not been refreshed yet and reports an incorrect agent status. When the update agent action is selected on this agent, the process cannot start and no operator message is displayed.

To solve this problem, you have to check the agent status locally and restart the agent instance if needed. Then, you have to re-issue the update agent command.

No Monitor Operator Messages available when updating with Centralized agent update process in SSL mode

You are centrally updating dynamic agents and fault-tolerant agents from Dynamic Workload Console. An agent is in running status in the Dynamic Workload Console, but the update process does not complete and no operator message is displayed.

Cause and solution

When running the Centralized Agent Update from Version 95 Fix Pack 5 to the current version with the event processor configured in SSL mode, no Monitor Operator Messages are displayed for either fault-tolerant agents and dynamic agents.

To solve this problem, perform one of the following steps:

- Perform centralized update from Version 95 Fix Pack 5 to Version 10 in SSL mode.
- Disable SSL communication for the event processor before running Centralized agent update, as follows:
    1. Stop WebSphere Application Server Liberty Base.
    2. Use the **eventProcessorEIFPort** optman option to define the port to be used for event processor communication. This automatically disables communication in SSL mode.
    3. Start WebSphere Application Server Liberty Base.
    4. Run JnextPlan to make the change effective.
    5. Perform Centralized agent update from Version 95 Fix Pack 5 to the current version.

Upgrading agents using HCL BigFix

Use the HCL BigFix Fixlets for IBM Workload Scheduler agents upgrade management to take advantage of:

- The HCL BigFix functions to view IBM Workload Scheduler information about all the agents installed on HCL BigFix endpoints.
- The Fixlets to automatically find all the IBM Workload Scheduler agents on which to install IBM Workload Scheduler upgrades. When the Fixlets become relevant, you can choose to schedule or run immediately an IBM Workload Scheduler upgrade installation.

HCL BigFix provides unified, real-time visibility and enforcement to deploy and manage upgrades to all endpoints from a single console.

Software requirements

Required software

You can use HCL BigFix Fixlets for IBM Workload Scheduler agents upgrade management in a distributed environment, by installing:

- IBM Workload Scheduler V9.3 Fix Pack 3 or later fault-tolerant agents, dynamic agents, IBM Z Workload Scheduler Agents.
- HCL BigFix for Lifecycle Management.

Upgrading remarks

Before you begin to upgrade agents using HCL BigFix, consider the following items:

- Make sure that you have at least 2 GB of free space under the root directory or filesystem (depending on your operating system).
- If on an agent there is more than one IBM Workload Scheduler instance, more than one baseline or Fixlet might be relevant for that agent. Make sure that you apply the baseline or Fixlet in the correct order and that you wait for an action to complete before starting a new one, because only one single action can be taken on the same agent at the same time.
- If there is more than one IBM Workload Scheduler instance installed on an agent; when you run a Fixlet to upgrade to a later level, this upgrade is made on one instance at a time, starting with the first one listed in the IBM Workload Scheduler registry. You cannot select a specific agent.

Creating and subscribing to the IBM Workload Automation Custom Site

The site hosts the IBM Workload Automation Fixlets that are pertinent to your network. To create and subscribe all the computers to the IBM Workload Automation custom site by using the HCL BigFix Console, perform the following procedure:

1. Select **Tools >Create Custom** Site.
2. You are prompted for a name for your custom site. Enter a name, for example, IBM Workload Automation, and click **OK**.
3. Select **All computers** to subscribe all the computers in the HCL BigFix environment to the IBM Workload Automation site.
4. From the **All Content Domain** panel, click the IBM Workload Automation site under **Sites ->Custom** to create the site.
5. From the **Details** tab, enter a description of the site. From the **Domain** pull-down menu, select a Domain to house your site.
6. From the **Computer Subscriptions** tab, indicate which subset of your HCL BigFix computers you want to subscribe to this site. For example **All Computers**.
7. From the **Operator Permissions** tab, you grant specific access permissions to specific operators.
8. Click **Save Changes** on the work area to complete the description of the site. You must enter your password to propagate your new custom site.

Importing IBM Workload Automation fixlets on the IBM Workload Automation Custom Site

To import the IBM Workload Automation Fixlets on the IBM Workload Automation Custom Site you created, use the HCL BigFix Console by performing the following procedure:

1. Select **File ->Import**. Using the **Import** dialog you import the `.bes` files, containing all the IBM® Workload Scheduler Fixlets that you downloaded from the BigFix.me community web site
2. After you click on **Import**, the **Import Content** dialog is displayed. Using it you review each HCL BigFix object to import (contained in .bse files) and to choose the site where to create those object. Choose the IBM Workload Automation Custom site as the site where to create the objects.
3. Click **OK** at the bottom of the **Import Content** dialog box to import the objects on the site.
4. From the navigation tree in the **All Content domain Panel**, click the icons labeled **Sites ->Custom Sites** ->IBM Workload Automation to review the imported Fixlets.

Customizing HCL BigFix to manage IBM Workload Scheduler agent upgrades

To customize HCL BigFix to manage an IBM Workload Scheduler agent upgrade, perform the following steps:

1. Open the HCL BigFix Console.
2. Log in to the HCL BigFix server by using the administrative credentials and perform the steps listed in the next sections to configure and customize the HCL BigFix environment to automate the IBM Workload Scheduler upgrade installation.

Enabling and subscribing to the Software Distribution external site

To enable and subscribe all the computers to the Software Distribution site using the HCL BigFix Console, perform the following steps:

1. Open the BigFix Management domain and scroll to the top to view the associated dashboards.
2. From the **License Overview** Dashboard, expand the **Lifecycle Management**, click **Software Distribution** hyperlink in the table of enabled sites.
3. Select the **Computer Subscriptions** tab.
4. Select **All computers** to subscribe all the computers in the HCL BigFix environment to the Software Distribution site.
5. Click **Save Changes** to save the subscription settings.

Installing and registering the Download Plug-in for Software Distribution

To install and register the Download Plug-in for Software Distribution using the HCL BigFix Console, perform the following steps:

1. From the navigation tree in the All Content domain, click **Sites**->**External Sites**->**Software Distribution**->**Fixlets and Tasks**.
2. From the resulting list panel on the right, click the **HCL BigFix Server: Install HCL BigFix Upload Maintenance Service for Software Distribution** Fixlet to open it. Ensure that the **Description** tab is selected.
3. From the **Description** tab, click the link or button corresponding to the Fixlet action. The **Take Action** dialog box is displayed.
4. If needed, you can refine the action settings using the appropriate tabs.
5. Click **OK** at the bottom of the **Take Action** dialog box to propagate the action to all the computers listed in this dialog box.
6. Repeat the procedure for the Fixlet: **HCL BigFix Server: Register Download Plug-in for Software Distribution**.

Uploading the IBM Workload Scheduler eImages and tools on the HCL BigFix server

To upload the IBM Workload Scheduler product eImages and the tools to unpack and deploy the product on the HCL BigFix server using the HCL BigFix Console, perform the following steps:

1. Download the installation images from IBM Fix Central.
2. In the navigation tree of the Systems Lifecycle domain panel, click **Software Distribution** ->**Manage Software Distribution Packages**.
3. From the resulting **Package Library** list panel on the right, click **New Package** to create the package for the IBM Workload Scheduler eImages and the package for the tools. Using the same panel, you must customize all the properties for these packages.
4. In the **Manage Files** tab at the bottom, click **Add Files** to upload the IBM Workload Scheduler eImages on the HCL BigFix server, one file at a time.
5. From the **Package Library** list panel, add the IBM Workload Scheduler tools package.
6. In the **Manage Files** tab at the bottom, click **Add Files** to upload the IBM Workload Scheduler tools on the HCL BigFix server, one file at a time.

Note: You must add the extract tools for every platform that you need. The extract tools are located in the IBM Workload Scheduler utility tools Multiplatform eImage that you downloaded from Passport Advantage. The following naming convention, specific for each operating system, was used:

- unzip-aix
- unzip-linux_s390
- unzip-linux_x86
- unzip-windows.exe

Using HCL BigFix relevant Fixlets to upgrade IBM Workload Scheduler agents

Fixlets and tasks are central to HCL BigFix. Using Relevance statements, they target specific computers, remediating only those HCL BigFix clients affected by an issue. They are both packaged with an action script that can resolve the issue with a simple mouse-click.

For example, IBM Workload Scheduler Fixlets find, if relevant, only the IBM Workload Scheduler agents that have installed a version earlier than 10.2.3. The related actions then prepare the instance to install the upgrade and then upgrade the agent.

Fixlets and tasks differ mainly on how they get resolved.

A Fixlet is triggered by a Relevance clause that detects a vulnerability, for example a version earlier than 10.2.3 applied to agents. When an action is invoked to solve the vulnerability, this Fixlet automatically loses relevance and is no longer applicable on that specific HCL BigFix client. When a Fixlet action propagates through your network, you can track its progress using the Console, Web Reports, and the Visualization Tool. When you remedy every HCL BigFix client in your network, the Fixlet is no longer relevant and is removed from the list. If the vulnerability returns, the Fixlet is shown again in the list to address the vulnerability again.

A task comes with one or more action scripts that help you to adjust settings or to run maintenance tasks.

At any time, you can open a Fixlet to inspect the underlying Relevance expressions that are used to target clients, as well as the action scripts that are designed to address the issue. The language used is close to the human language to give you a high degree of confidence in both applicability and efficacy of the remedial action. You can also see precisely which computers in your network are affected by each Fixlet. When propagated, you can view the progress and ultimate history of each action taken on a client basis.

IBM Workload Scheduler provides the following Fixlets for each operating system to upgrade agents to the new version:

1. **Prepare the upgrade of the IBM Workload Scheduler *type_of_agent* agent to version 10.2.3 for *platform***
2. **Install the IBM Workload Scheduler *type_of_agent* agent to version 10.2.3 for *platform***

Where *type_of_agent* can be fault-tolerant, dynamic, for z/OS and *platform* is one of the supported operating systems.

If the first Fixlet is relevant and you click **Take Action**, HCL BigFix prepares the IBM Workload Scheduler agent for the upgrade by performing the following steps:

- Downloads the images from the HCL BigFix server or relay.
- Extracts the images.
- Checks if the IBM Workload Scheduler command line tools are running (**`conman`**, **`composer`**, **`fileaid`**). If they are running, the action fails.
- Enables the Install Fixlet for the upgrade

If one of the actions fails, the Fixlet fails and remains relevant. You can check the failed action by using the **Status** tab of the action. Perform the necessary steps to solve the problems on the agents and rerun the action.

Note: If the extract step fails, check if the extract tool is present on the agent. If it is not present, install the extract tool and rerun the action.

Note: If the procedure to prepare the agent upgrade fails with the following error:

```
Completed // Delete $TMP/run.sh
Completed delete "{parameter "TMP"}/run.sh"
Completed // Move __createfile to $TMP/run.sh
Completed move __createfile "{parameter "TMP"}/run.sh"
Completed // Execute run.sh
Completed wait sh "{parameter "TMP"}/run.sh"
Completed // Continue if the return code of the previous command was 0
Failed continue if {exit code of action = 0}
```

the problem is caused by an IBM Workload Scheduler process that did not stop. To solve the problem, run the following actions:

1. On UNIX operating systems, check the file:

   ```
   /tmp/TWA/tws952/tws952_process_agent_user.txt
   ```

   to find information about the process that is still running.
2. Kill the process.
3. Rerun the action.

To find information about the log file location for HCL BigFix on several operating systems, see: HCL BigFix Common File Locations .

If all the actions succeed, the Fixlet is no longer relevant and the next Fixlet becomes relevant. If you click **Take Action** for the new one, it upgrades the previously prepared agent instance to 10.2.3, performing the following steps:

- Upgrades the instance.
- Resets the fence to the original value.
- Links back to the domain manager.

Also in this case you can check the status of the action through the relative tab and, in case of errors, solve the problems and rerun the action until it succeeds.

Displaying relevant IBM Workload Scheduler Fixlets

To display an IBM Workload Scheduler Fixlet using the HCL BigFix Console, perform the following procedure:

1. From the navigation tree in the **Domain** Panel, click the icon labeled **Fixlets and Tasks**. The list panel is displayed on the right.
2. From the list panel, click any IBM Workload Scheduler Fixlet to open it. The body of the Fixlet message is displayed in the work area.
3. Each Fixlet contains a work area with the following four tabs:

   Description

   This page provides a descriptive explanation of the problem and one or more actions to fix it. The actions are represented by links at the bottom of the description page. Click an action to open the **Take Action** dialog, to choose other targets, or to schedule the action. If you click by mistake an action hyperlink before the actual deployment, you always have the chance to modify or cancel the action.

   Details

   This dialog contains the Fixlet and task properties such as category, security ID, download size, source, severity, and date. It also lists the code behind the Relevance expressions and the actions. In a text box at the bottom of this dialog, you can type a comment that remains attached to this item.

   Applicable Computers

   This is a list of all the computers targeted by the selected Fixlet or task. You can filter the list by selecting items from the folders on the left, and sort the list by clicking the column headers.

   Action History

   This is a list of actions that have been deployed by this Fixlet or task. If this item is new, the list is empty. You can filter the actions using the left panel, and sort them by clicking the column headers above the right-hand list.

Deploying IBM Workload Scheduler actions

To deploy an IBM Workload Scheduler action using the HCL BigFix Console, perform the following procedure:

1. Click the list panel to open a relevant Fixlet or task. Make sure the **Description** tab is selected.
2. Read the description carefully. Scroll down to see the suggested actions.
3. Click the **Details** tab and search the action. Examine the Relevance section and the action script itself.

4. In the **Description** tab, click the link corresponding to the Fixlet action or click the **Take Action** button.
5. The **Action Parameter** pop-up window is displayed. Provide the required information. Click **OK**.
6. The **Take Action** dialog box is displayed. In the **Preset** pull-down menu, you can accept the default settings or select **Policy** to set an action with no expiration date. For more information about presets, see the section about **Custom Actions**.
    a. You can refine the list of targeted computers using the **Target** tab. Use the computer tree in the left panel to filter the list of workstations in the right panel.
    b. In the **Execution** tab, you can set various scheduling constraints and behaviors.
    c. In the **Messages** tab, you can create an optional message to be shown on the HCL BigFix client computers.
    d. In the **Action Script** tab, operators with Custom Authoring permissions can modify the action script.
    e. Use the other interface tabs to further modify the Action settings.
7. Click **OK**

Note: If you are taking an action that applies to different computers, when you are prompted to insert values for the action parameters, you must leave the default values; you must not specify other values.

The action is propagated to all the computers targeted in the **Take Action** dialog. After the action ends successfully and the targeted computers are fixed, those computers no longer report this Fixlet as relevant.

Monitoring IBM Workload Scheduler actions

When you decide to take a proposed action, you have several deployment options. For example, you might schedule the action to run unattended after midnight or to run with user involvement during the day.

After you schedule the actions, the HCL BigFix server attempts to identify the computers suitable for those actions. Ideally, the HCL BigFix client gathers the action information from the action site and performs it immediately. However, some computers might be powered off and others might be mobile devices undocked when the action is deployed. As soon as these computers become available, the remedial action is applied.

To monitor a deployed action, using the HCL BigFix Console, click the **Actions** icon in the Domain panel navigation tree.

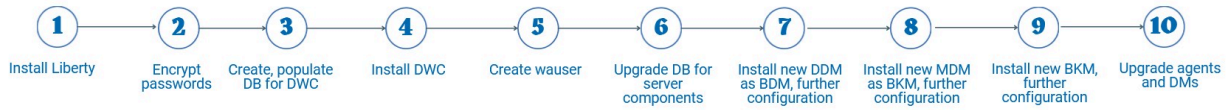
If you have not yet deployed an action or all the actions completed, this list is empty. Otherwise, click any action to view its status, whether it is evaluating, waiting, running, fixed, or failed. You can also add comments to the action.

Actions might go through several states as they are collected, evaluated, and run by clients.

Note: If an action failed for any reason and its state is Open, before running it again, make sure to stop it and that it is not listed in the actions list.

Parallel upgrade from version 9.4.0.*x* to version 10.2.3

Detailed steps to perform a parallel upgrade from version 9.4.0.*x* to version 10.2.3



A number of major product improvements have been inserted starting from version 9.5. For this reason, when upgrading from version 9.4.0.*x*, you have to perform a fresh installation of the following components and prerequisites:

- WebSphere Application Server Liberty Base
- Dynamic Workload Console
- Dynamic Workload Console database
- master domain manager
- backup master domain manager

- dynamic domain manager
- backup dynamic domain manager

Before you start the upgrade, ensure you have performed the following procedures:

- Connecting the Dynamic Workload Console to a new node or database *(on page 161)*. If you are currently using Derby, install a supported database before exporting Dynamic Workload Console data. This is necessary because Derby is no longer supported as of version 10.2.3.
- Installing the fix for APAR IJ47731 on the master domain manager. The fix is available on IBM Fix Central. For more information, see IJ47731: CHECKSYNC JOB FAILED IN IWS 10.1 FP02 - AWSJCL075E ERROR
- Configuring TLS to the appropriate version *(on page 222)* on the 9.4 master domain manager to ensure communication in your environment.
- Converting default certificates *(on page 223)*, if you are using default certificates in your current environment. Use this procedure to convert the certificates from the JKS to the PEM format, then copy them to the workstations where you plan to install the server components (dynamic domain manager and its backups, master domain manager and its backups) and the Dynamic Workload Console.

  Certificates are now required when installing or upgrading IBM® Workload Scheduler. You can no longer install nor upgrade IBM® Workload Scheduler without securing your environment with certificates. The required certificates are:
  - ca.crt
  - tls.key
  - tls.crt

  For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).
- Install the latest fix pack for version 9.4 on all workstations in your environment.

To perform a parallel upgrade from version 9.4.0.*x* to 10.2.3, perform the following steps:

1. Installing WebSphere Application Server Liberty *(on page 225)* on the workstations hosting the Dynamic Workload Console and the server components. This is a prerequisite component which replaces WebSphere Application Server used in version 9.4.0*x*.
2. Encrypting passwords (optional) *(on page 226)*
3. Creating and populating the database for the Dynamic Workload Console *(on page 227)*
4. Installing the Dynamic Workload Console *(on page 236)*
5. Creating the IBM Workload Scheduler administrative user *(on page 239)* on the workstations which will host the components at 10.2.3 level.
6. Upgrading the database for the server components *(on page 239)*
7. Installing a new dynamic domain manager configured as a backup *(on page 242)*
   a. Installing a new dynamic domain manager configured as a backup *(on page 242)*
   b. Switching the dynamic domain manager to the new dynamic domain manager configured as backup *(on page 244)*
   c. Installing a new backup dynamic domain manager *(on page 245)* to replace the backup dynamic domain manager which you have switched to become the current dynamic domain manager.
   d. Switching back to the old dynamic domain manager (optional) *(on page 247)*
8. Installing the new master domain manager configured as a backup *(on page 247)*
   a. Ensuring communication in your environment *(on page 259)*
   b. Switching the master domain manager to the new backup master *(on page 252)*
   c. Making the switch manager permanent *(on page 253)*
   d. Customizing and submitting the optional FINAL job stream *(on page 254)*
9. Installing a new backup master domain manager *(on page 255)*
   a. Installing a new backup master domain manager *(on page 256)*
   b. Uninstalling the back-level backup master domain manager *(on page 260)*
10. Upgrading agents and domain managers *(on page 261)*

Configuring TLS to the appropriate version

Transport Layer Security (TLS) is a cryptographic protocol designed to provide secure communication over a computer network. It ensures that data transmitted between applications, such as web browsers and servers, remains private and tamper-proof. Setting TLS to version 1.2 is required to ensure communication between 9.4 and 10.2.3 components.

In back-level environments, for example 9.4, SSL is not enabled by default and TLS version 1.2 needs to be enabled on the back-level master domain manager to enable communication. Perform the following steps on the back-level master domain manager:

1. Browse to the `<JazzSMHome>/profile/config/cells/JazzSMNode01Cell` path, where

   <JazzSMHome>

   is the directory where Jazz for Service Management is installed.
2. Open the `security.xml` file in a flat-text editor.
3. Change the value of the **sslProtocol** parameter to **TLSv1.2** and save the file.
4. Browse to the *JazzSM*`/profile/properties` path.
5. Open the `ssl.client.props` file in a flat-text editor.
6. Change the **com.ibm.ssl.protocol** parameter to `TLSv1.2` and save the file.
7. Run the following commands from the TWA_home`/wastools` directory to stop and restart the master domain manager:

   ```
   ./ stopWas.sh -direct -\user| wauser -password \password
   ./ startWas.sh -direct
   ```
8. Run the following commands from the *DWC_home*`/wastools` directory to stop and restart the Dynamic Workload Console:

   ```
   ./ stopWas.sh -direct -\user| DWCuser -password \password
   ./ startWas.sh -direct
   ```

For more information, see Switching from SSLv3 to TLSv1.2 and steps 2 and 3 in How to Run Composer on a 9.5 FTA Connecting to a 9.4 MDM

Converting default certificates

Procedure to extract and convert default certificates generated in your current version prior to upgrading.

If you are using default certificates, extract and convert them before you start the upgrade. Perform the following steps:

1. Set the IBM® Workload Scheduler environment, as described in Setting the environment variables *(on page 138)*.
2. To ensure the keytool and openssl commands start correctly on all operating systems, browse to the folder where the keytool and openssl commands are located and launch the commands as follows:

   ```
   cd <TWS_DIR>/JavaExt/jre/jre/bin
   ```

   ```
   ./keytool -importkeystore -srckeystore TWSServerKeyFile.jks -destkeystore
   <path_of_extracted_certs>/server.p12 -deststoretype pkcs12
   ```

   ```
   cd <TWS_DIR>/tmpOpenSSL64/1.1/bin/openssl
   ```

   ```
   ./openssl pkcs12 -in <path_of_extracted_certs>/server.p12 -out
   <path_of_extracted_certs>/tls.tot
   ```

   The location of the `TWSServerKeyFile.jks` varies depending on the IBM® Workload Scheduler version you have currently installed, as follows:

   versions 9.5 and later

   TWA_DATA_DIR`/usr/servers/engineServer/resources/security`

   versions 9.4 and earlier

   TWA_home`/WAS/TWSProfile/etc`
3. Open the `tls.tot` file with any text editor.
4. From the `tls.tot` file, copy the private key to a new file named `tls.key`.
   The `tls.key` file must be structured as follows:

   ```
   ----BEGIN ENCRYPTED PRIVATE KEY----
   <private_key>
   ----END ENCRYPTED PRIVATE KEY----
   ```

Note: Insert a carriage return after each key, so that an empty line is inserted after each key.

5. From the `tls.tot` file, copy the public key to a new file named `tls.crt`.
   The `tls.crt` file must be structured as follows:

```
----BEGIN CERTIFICATE----
<public_key>
----END CERTIFICATE----
```

Note: Insert a carriage return after each key, so that an empty line is inserted after each key.

6. Copy the contents of the `tls.crt` file into a new file named `ca.crt`. If you want to upgrade a dynamic domain manager, also copy the contents of the `tls.crt` file into another new file named `jwt.crt`.
7. Create a file named `tls.sth` containing the passphrase you have specified for creating the `.p12` certificate in step 2 *(on page 223)*, encoded in `base64` format. To create the `tls.sth` file, use the following command:

```
./secure -password your_password -base64 e -out
<path_of_extracted_certs>/tls.sth
```

   If you are using a version earlier than 10.*x*, you can find the secure script in the installation package of the 10.2.3 version you are upgrading to. You can launch the script from on of the following paths:

   master domain manager and agent

```
<10.2.3_extracted_image_dir>/TWS/<interp>/Tivoli_LWA_<interp>/TWS/bin
```

   Dynamic Workload Console

```
<10.2.3_extracted_image_dir>/DWC/<interp>/bin
```

   where

   <interp>

   is the operating system you are installing on

   As an alternative, you can use the following command on UNIX workstations:

```
echo -n "passwordToEncode" | base64 >> tls.sth
```

8. Browse to the GSKit folder and extract the client certificates from the *TWA_DATA_DIR*`/ssl/GSKit` folder by running the following commands, depending on the IBM® Workload Scheduler version you have currently installed:

```
cd <TWS_DIR>/tmpGSKit64/8/bin
```

   versions 9.5 and later

```
./gsk8capicmd_64 -cert -extract -db <TWA_DATA_DIR>/ssl/GSKit/TWSClientKeyStore.kdb
-stashed -label client -target <path_of_extracted_certs>/client.crt
```

   versions 9.4 and earlier

```
./gsk8capicmd_64 -cert -extract -db <TWS_DIR>/ssl/GSKit/TWSClientKeyStore.kdb
-stashed -label client -target <path_of_extracted_certs>/client.crt
```

9. Create a folder named `additionalCAs` in the folder where you extracted the certificates and move the `client.crt` file created in step 8 *(on page 224)* to the `additionalCAs` folder.
10. Insert the `client.crt` in the `additionalCAs` folder when providing the certificates to the installation script with the **sslkeysfolder** parameter.
11. Assign the correct permissions (755) and ownerships to extracted certificates, as follows:

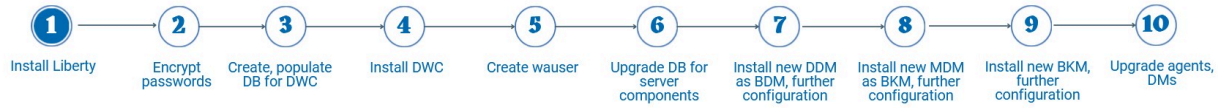
```
chmod -R 755 <path_of_extracted_certs>
```

You have now extracted and converted your certificates for use with version 10.2.3.

You can now upgrade WebSphere Application Server Liberty, as described inInstalling WebSphere Application Server Liberty *(on page 225)*. When upgrading IBM® Workload Scheduler components in upcoming steps, provide the path to the folder where you extracted the certificates using the **sslkeysfolder** parameter when running the installation scripts. For more information about the installation scripts, see Reference *(on page 300)*.

Installing WebSphere Application Server Liberty

Installing WebSphere Application Server Liberty to the latest supported version. This is an optional step you might want to perform before you upgrade the Dynamic Workload Console and the server components.



On AIX and Linux workstations, ensure you permanently set the **ulimit** parameter as follows:

- data segment process (option **-d**) = `unlimited`
- file size (option **-f**) = `unlimited`
- max user processes (option **-u**) = >`260000` up to `unlimited`
- open files (option **-n**) = >`100000` up to `unlimited`
- max memory size (option **-m**) = `unlimited`
- stack size (option **-s**) = >`33000` up to `unlimited`

On the master domain manager, these settings must be applied to:

- `root`
- the IBM® Workload Scheduler administrative user

On the Dynamic Workload Console, these settings must be applied to:

- `root`
- the Dynamic Workload Console installation user (if this user is different from root)

Ensure that your system meets the operating system and Java requirements. For more information, see WebSphere Application Server Liberty Base detailed system requirements.

You can quickly install WebSphere Application Server Liberty Base by extracting an archive file on all supported platforms.

Install WebSphere Application Server Liberty Base on all of the following workstations, which comprise a typical installation:

- master domain manager
- backup domain manager
- two Dynamic Workload Console installations on two separate workstations

If you plan to install a dynamic domain manager and its backup, these components require a separate WebSphere Application Server Liberty Base installation.

To extract the archive, you can use your own Java Ext or use the Java Ext provided with the IBM® Workload Scheduler image. The provided Java Ext is located in the following path in the image for your operating system: `<IMAGE_DIR>/TWS/<INTERP>/Tivoli_Eclipse_<INTERP>/TWS/JavaExt/`.

To install WebSphere Application Server Liberty Base, perform the following steps:

1. Find out which version of WebSphere Application Server Liberty Base is required, by running the Detailed Software Requirements report and browsing to the **Prerequisites** tab.
2. Download WebSphere Application Server Liberty Base from Recommended updates for WebSphere Application Server Liberty.

3. Install WebSphere Application Server Liberty Base by extracting the archive file to a directory of your choice.

   On Windows operating systems

   ```
   java -jar <liberty_download_dir>\wlp-base-all-<version>.jar
   --acceptLicense <install_dir>
   ```

   On UNIX operating systems

   ```
   ./java -jar <liberty_download_dir>/wlp-base-all-<version>.jar
   --acceptLicense <install_dir>
   ```

   where:

   <liberty_download_dir>

   The directory where you downloaded WebSphere Application Server Liberty Base.

   <version>

   The number of the version.

   <install_dir>

   The directory where you want to install WebSphere Application Server Liberty Base.

   Note: Note that the value of the *<install_dir>* parameter must match the value to be defined for the **wlpdir** parameter when installing the master domain manager and its backup, dynamic domain manager and its backup, and the Dynamic Workload Console.

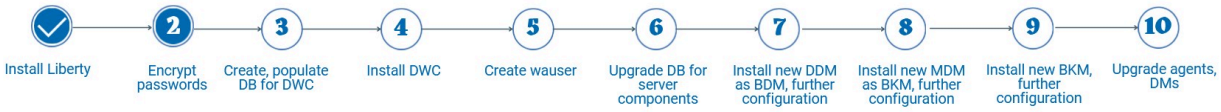
4. Ensure the IBM® Workload Scheduler administrative user has the rights to run WebSphere Application Server Liberty Base and full access to the installation directory. If WebSphere Application Server Liberty Base is shared between the master domain manager and the Dynamic Workload Console, ensure also the Dynamic Workload Console user has the same rights.

You have now successfully installed WebSphere Application Server Liberty Base.

You can now proceed to Encrypting passwords (optional) *(on page 226)* or to Creating and populating the database for the Dynamic Workload Console *(on page 227)*.

Encrypting passwords (optional)

How to encrypt passwords required by the installation process



You can optionally encrypt the passwords that you will use while installing, upgrading, and managing IBM® Workload Scheduler. The secure command uses the AES method and prints the encrypted password to the screen or saves it to a file.

Note: It is important you understand the limits to the protection that this method provides. The custom passphrase you use to encrypt the passwords is stored in clear format in the `passphrase_variables.xml` file, stored in `configureDropin`. To fully understand the implications of this method, it is recommended you read the information provided by WebSphere Application Server Liberty Base at the link Liberty: The limits to protection through password encryption.

You can perform a typical procedure, which uses a custom passphrase, as described in the following scenario. For more information about all secure arguments and default values, see Optional password encryption - secure script *(on page 300)*.

Encrytping the password

1. Browse to the folder where the secure command is located:
   - Before the installation, the command is located in the product image directory, `<image_directory>/TWS/<op_sys>/Tivoli_LWA_<op_sys>/TWS/bin`
   - After the installation, the command is located in `TWA_home/TWS/bin`
2. Depending on your operating system, encrypt the password as follows:

   Windows operating systems

   ```
   secure -password password -passphrase passphrase
   ```

   UNIX operating systems

   ```
   ./secure -password password -passphrase passphrase
   ```

   z/OS operating systems

   ```
   ./secure -password password -passphrase passphrase
   ```

   where

   -password

   Specifies the password to be encrypted.

   -passphrase

   Optional. Specifies the custom passphrase that is used to generate the key with which the command encrypts the password. If you set this parameter, inform the user who installs IBM Workload Automation that they must define the **SECUREWRAP_PASSPHRASE** environment variable in the same shell from which they run the installation command, and set it to the same value as the **passphrase** argument. On Windows operating systems, the passphrase must be at least 8 characters long.
3. Provide both the encrypted password and custom passphrase to the user in charge of installing IBM Workload Automation. You can use encrypted passwords only in association with the specific passphrase used to encrypt them.

Installing with the encrypted password

The user in charge of installing IBM Workload Automation must set the **SECUREWRAP_PASSPHRASE** environment variable by performing the following steps:

1. Open a brand new shell session.
2. Ensure that no value is set for the **SECUREWRAP_PASSPHRASE** environment variable.
3. Define the **SECUREWRAP_PASSPHRASE** environment variable and set it to the passphrase defined by the user who ran the secure command, as follows:

   ```
   SECUREWRAP_PASSPHRASE=<passphrase>
   ```

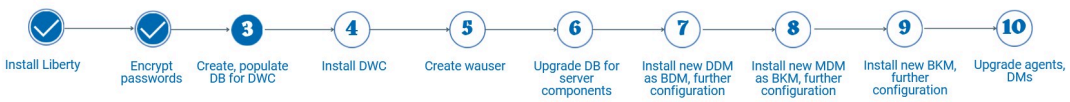
   You can use encrypted passwords only in association with the specific passphrase used to encrypt them.
4. In the same shell session, provide the encrypted passwords when running any command that uses a password. An encrypted password looks like the following example:

   ```
   {aes}AFC3jj9cROYyqR+3CONBzVi8deLb2Bossb9GGroh8UmDPGikIkzXZzid3nzY0IhnSg=
   ```

You can now proceed to Creating and populating the database for the Dynamic Workload Console *(on page 227)*.

Creating and populating the database for the Dynamic Workload Console

Create and populate the database for the Dynamic Workload Console



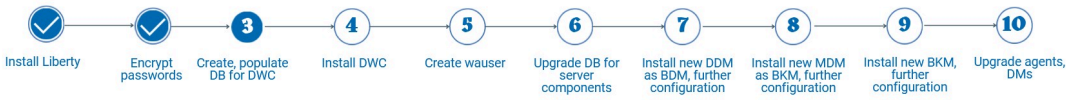
If you are currently using Derby, you need to install a supported database and migrate your data. This is necessary because Derby is no longer supported as of version 10.2.3. For more information, see Connecting the Dynamic Workload Console to a new node or database *(on page 161)*.

If you are using a database other than Derby, create and populate the database tables for the Dynamic Workload Console by following the procedure appropriate for your RDBMS:

Creating and populating the database for DB2 for the Dynamic Workload Console

Instructions for creating and populating the Dynamic Workload Console database for DB2

Ensure a DB2 database is installed.



You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in FAQ - Database customizations *(on page 60)*.

DB2 requires a specific procedure in which you first create the database and then create and populate the database tables. To simplify the database creation, a customized SQL file named `create_database.sql` is provided containing the specifics for creating the Dynamic Workload Console database. The database administrator can use this file to create the database. After the database has been created, you can proceed to create and populate the database tables.

You can optionally configure DB2 in SSL mode on UNIX operating systems by specifying the **sslkeysfolder** and **sslpassword** parameters when you run the configureDb command. For more information, see How can I use certificates when Db2 or PostgreSQL is in SSL mode? *(on page 66)*.

You can run the configureDb command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

For more information about all parameters and supported values of the configureDb command, see Database configuration - configureDb script *(on page 301)*.

Default values are stored in the `configureDb.properties` file, located in *`image_location`*. If you need to modify any of the default values, edit the `configureDb.properties` file, but do not modify the `configureDb.template` file located in the same path.

To create and populate the Dynamic Workload Console database and schema for DB2, perform the following steps:

1. On the workstation where you plan to install the Dynamic Workload Console, extract the Dynamic Workload Console package to a directory of your choice.
2. Browse to the *`image_location`*`/DWC_`*`interp_name`*`/tools` path.
3. Edit the `create_database.sql` file by replacing the default value for the database name (**DWC**) with the name you intend to use.
4. Provide the `create_database.sql` file to the DB2 administrator to run on the DB2 database.
   The following command creates the Dynamic Workload Console database:

   ```
   db2 -tvf file_location>/create_database.sql
   ```

5. Instruct the DB2 administrator to create the DB2 user on the server hosting the DB2 database. You will then specify this user with the `dbuser` parameter when creating and populating the database with the configureDb command on the

Dynamic Workload Console. When you run the configureDb command, this user is automatically granted access to the Dynamic Workload Console tables on the database server.

6. On the server where you plan to install the Dynamic Workload Console,browse to *image_location*/DWC_*interp_name*.
7. Type the following command to create and populate the Dynamic Workload Console database tables with typical settings:

   On Windows operating systems

   ```
   cscript configureDb.vbs --rdbmstype DB2 --dbhostname DB_hostname
           --dbport db_port --dbname db_name --dbuser db_user
           --dbadminuser DB_admin_user --dbadminuserpw DB_admin_pwd
   ```

   On UNIX operating systems

   ```
   ./configureDb.sh --rdbmstype DB2 --dbhostname DB_hostname
           --dbport db_port --dbname db_name --dbuser db_user
           --dbadminuser DB_admin_user --dbadminuserpw DB_admin_pwd
   ```

   where:

   --rdbmstype

   The database vendor.

   --dbhostname *db_hostname*

   The host name or IP address of database server.

   --dbport *db_port*

   The port of the database server.

   --dbname *db_name*

   The name of the Dynamic Workload Console database.

   --dbuser *db_user*

   The database user you must create before running the configureDb command. When you run the configureDb command, this user is automatically granted access to the Dynamic Workload Console tables on the database server.

   --dbadminuser *db_admin_user*

   The database administrator user that creates the Dynamic Workload Console schema objects on the database server.

   --dbadminuserpw *db_admin_password*

   The password of the DB administrator user that creates the Dynamic Workload Console schema objects on the database server. Special characters are not supported.

   Note: The following parameters specified with the configureDb command are also required when installing theDynamic Workload Console and their values must be the same:
   - **--rdbmstype**
   - **--dbhostname**
   - **--dbport**
   - **--dbname**
   - **--dbuser**

You have now successfully created and populated the Dynamic Workload Console database.

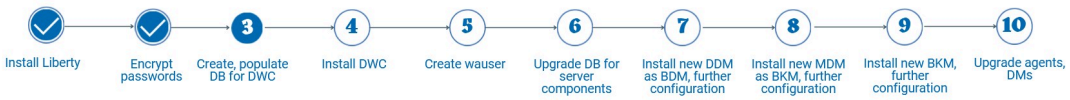
For more information about all parameters and supported values of the configureDb command, see Database configuration - configureDb script *(on page 301)*.

You can now proceed to Installing the Dynamic Workload Console *(on page 236)*.

Creating and populating the database for DB2 for z/OS for the Dynamic Workload Console

Instructions for creating and populating the database for DB2 for z/OS for Dynamic Workload Console

Ensure a DB2 for z/OS database is installed.



You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in FAQ - Database customizations *(on page 60)*.

DB2 for z/OS requires a specific procedure in which you first create the database and then create and populate the database tables. To simplify the database creation, a sample JCL named `EQQINDWC` is provided with APAR PH22448 containing the specifics for creating the Dynamic Workload Console database. The database administrator can use this file to create the database. After the database has been created, you can proceed to create and populate the database tables.

You can run the **configureDb** command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

For more information about all parameters and supported values of the `configureDb` command, see Database configuration - configureDb script *(on page 301)*.

If you need to modify any of the default values, edit the `configureDb<database_vendor>.properties` file, but do not modify the `configureDb<database_vendor>.template` file located in the same path.

To create and populate the Dynamic Workload Console database and schema for DB2 for z/OS, perform the following steps:

1. From the SEQQSAMP library, edit the `EQQINDWC` sample JCL as required.

   Note: The `EQQINDWC` sample JCL is provided with the APAR PH22448. If you did not install this APAR, create a JCL named `EQQINDWC` that looks like the following example:

```
//JOBCARD
//*********************************************************************/
//*                                                                   */
//*  SECURITY CLASSIFICATION:                                         */
//*  Licensed Materials - Property of HCL 5698-T08                    */
//*  Copyright HCL Technologies Ltd. 2020 All rights reserved.        */
//*  US Government Users Restricted Rights - Use, duplication         */
//*  or disclosure restricted by GSA ADP Schedule Contract            */
//*                                                                   */
//* CREATES DB2 STORAGE GROUP AND DATABASE for DWC                    */
//* NOTE1:You must tailor this JCL sample to conform to               */
//*       installation standards defined at your location.            */
//*       - Add a JOB card                                            */
//*       - Change following DB/2 values according to your            */
//*         current environment:                                      */
//*         - DSN.V11R1M0.SDSNLOAD      DB/2 library                  */
//*         - DSN111.RUNLIB.LOAD        DB/2 run library              */
//*         - DBB1                      DB/2 system name              */
//*         - DSNTIA11                  DB/2 DSNTIAD plan name        */
//*         - volname                   volume name                   */
//*         - catname                   catalog name                  */
//*       - Change all the occurrences of                             */
//*         TWSSDWC if you need a storage group with a different name*/
//*                                                                   */
//* Flag Reason   Rlse   Date   Origin Flag Description               */
//* ---- -------- ------ ------ ------ ----------------------------  */
//* $EGE=PH22448  950    200121 ZLIB: DB2 on zLiberty                 */
//* $ETA=PH53936  101 220418 MR: EQQINDWC MEMBER OF SEQQSAMP FOR      */
//*                              CREATION OF DB2 DATABASE FOR         */
//*                              DWCFAILS FOR DB2 V12R1M504 OR        */
//*                              higher levels                        */
//*********************************************************************/
```

```
//EQQINDWC EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB  DD  DISP=SHR,DSN=DSN.V11R1M0.SDSNLOAD
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  *
  DSN SYSTEM(DBB1)
  RUN PROGRAM(DSNTIAD) PLAN(DSNTIA11) LIB('DSN111.RUNLIB.LOAD')
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSIN    DD *
SET CURRENT APPLICATION COMPATIBILITY = 'V10R1';
CREATE STOGROUP TWSSDWC VOLUMES(volname) VCAT catname;
CREATE DATABASE DWC
BUFFERPOOL BP0
INDEXBP BP16K0
STOGROUP TWSSDWC
CCSID UNICODE;
COMMIT;
```

2. Instruct the DB2 for z/OS administrator to create the DB2 for z/OS user on the server hosting the DB2 for z/OS database. You will then specify this user with the `dbuser` parameter when creating and populating the database with the `configureDb` command on the Dynamic Workload Console. When you run the `configureDb` command, this user is automatically granted access to the Dynamic Workload Console tables on the database server.
3. On the server where you plan to install the Dynamic Workload Console, browse to the directory where you extracted the Dynamic Workload Console image.
4. Type the following command to create and populate the Dynamic Workload Console database tables with typical settings:

   On Windows operating systems

   ```
   cscript configureDb.vbs --rdbmstype DB2Z --dbhostname DB_hostname
   --dbport db_port --dbname db_name --dbuser db_user
   --dbadminuser DB_admin_user --dbadminuserpw DB_admin_pwd
   --zlocationname zOS_location_containing_db --zbufferpoolname buffer_pool_in_zOS_location
   ```

   On UNIX operating systems

   ```
   ./configureDb.sh --rdbmstype DB2Z --dbhostname DB_hostname
   --dbport db_port --dbname db_name --dbuser db_user
   --dbadminuser DB_admin_user --dbadminuserpw DB_admin_pwd
   --zlocationname zOS_location_containing_db --zbufferpoolname buffer_pool_in_zOS_location
   ```

   On z/OS operating systems

   ```
   ./configureDb.sh --rdbmstype DB2Z --dbhostname DB_hostname
   --dbport db_port --dbname db_name --dbuser db_user
   --dbadminuser DB_admin_user --dbadminuserpw DB_admin_pwd
   --zlocationname zOS_location_containing_db --zbufferpoolname buffer_pool_in_zOS_location
   ```

   where:

   --rdbmstype

   The database vendor.

   --dbhostname *db_hostname*

   The host name or IP address of database server.

   --dbport *db_port*

   The port of the database server.

   --dbname *db_name*

   The name of the Dynamic Workload Console database.

   --dbuser *db_user*

   The database user you must create before running the `configureDb` command. When you run the `configureDb` command, this user is automatically granted access to the Dynamic Workload Console tables on the database server.

   --dbadminuser *db_admin_user*

The database administrator user that creates the Dynamic Workload Console schema objects on the database server.

--dbadminuserpw ***db_admin_password***

The password of the DB administrator user that creates the Dynamic Workload Console schema objects on the database server. Special characters are not supported.

--zlocationname ***zos_location_containing_db***

The name of an already existing location in the z/OS environment that will contain the new database. The default value is LOC1.

--zbufferpoolname ***buffer_pool_in zos_location***

The name of an already existing buffer pool created in the location specified by `-zlocationname`. The default value is BP32K.

Note: The following parameters specified with the **configureDb** command are also required when installing the Dynamic Workload Console and their values must be the same:

- **--rdbmstype**
- **--dbhostname**
- **--dbport**
- **--dbname**
- **--dbuser**
- **--zlocationname**

You have now successfully created and populated the Dynamic Workload Console database.

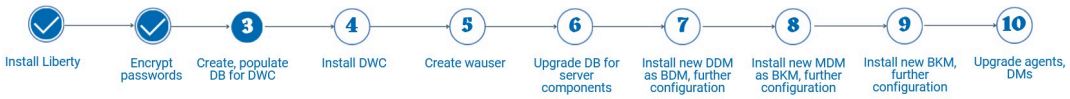
For more information about all parameters and supported values of the `configureDb` command, see Database configuration - configureDb script *(on page 301)*.

You can now proceed to Installing the Dynamic Workload Console *(on page 236)*.

Creating the database for Oracle and Amazon RDS for Oracle for the Dynamic Workload Console

Instructions for creating and populating the Dynamic Workload Console database for Oracle and Amazon RDS for Oracle

Ensure the required tablespace for Dynamic Workload Console data has been already created on the Oracle database server which hosts the Dynamic Workload Console database.



You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in FAQ - Database customizations *(on page 60)*.

You can run the configureDb command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

Default values are stored in the `configureDbOracle.properties` file, located in *`image_location`*. If you need to modify any of the default values, edit the `configureDbOracle.properties` file, but do not modify the `configureDbOracle.template` file located in the same path.

For more information about all parameters and supported values of the configureDb command, see Database configuration - configureDb script *(on page 301)*.

To create and populate the Dynamic Workload Console database, perform the following steps:

1. On the server where you plan to install the Dynamic Workload Console, extract the Dynamic Workload Console package to a directory of your choice.
2. Browse to the directory where you extracted the package.
3. Type the following command to populate the Dynamic Workload Console database with typical settings:

   On Windows operating systems

   ```
   cscript configureDb.vbs --rdbmstype ORACLE --dbname service_name
   --dbuser db_user --dbpassword DB_password --dbhostname DB_hostname
   --dbadminuser DB_administrator --dbadminuserpw DB_administrator_password
   --iwstsname USERS
   ```

   On UNIX operating systems

   ```
   ./configureDb.sh --rdbmstype ORACLE --dbname service_name
   --dbuser db_user --dbpassword DB_password --dbhostname DB_hostname
   --dbadminuser DB_administrator --dbadminuserpw DB_administrator_password
   --iwstsname USERS
   ```

   where:

   --rdbmstype

   The database vendor.

   --dbname *db_name*

   The service name of the Dynamic Workload Console database.

   —dbuser *db_user*

   The user to be granted access to the Dynamic Workload Console tables on the database server.

   --dbpassword *db_password*

   The password for the user that has been granted access to the Dynamic Workload Console tables on the database server. Special characters are not supported.

   --dbhostname *db_hostname*

   The host name or IP address of database server.

   --dbadminuser*db_admin_user*

   The database administrator user that creates the Dynamic Workload Console schema objects on the database server.

   --dbadminuserpw *db_admin_password*

   The password of the DB administrator user that creates the Dynamic Workload Console schema objects on the database server. Special characters are not supported.

   --iwstsname|-tn *table_space_name*

   The name of the tablespace for Dynamic Workload Console data. This parameter is required.

   Note: The following parameters specified with the configureDb command are also required when installing the Dynamic Workload Console and their values must be the same:
   - **rdbmstype**
   - **dbhostname**
   - **dbport**
   - **dbname**
   - **dbuser**
   - **dbpassword**

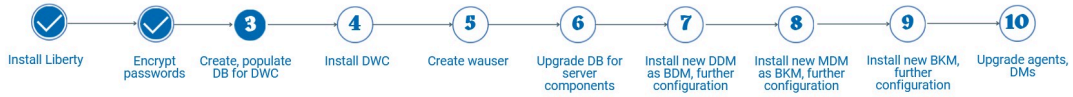
You have now successfully created and populated the Dynamic Workload Console database.

For more information about all parameters and supported values of the configureDb command, see Database configuration - configureDb script *(on page 301)*.

You can now proceed to Installing the Dynamic Workload Console *(on page 236)*.

Creating and populating the database for MSSQL for the Dynamic Workload Console

Instructions for creating and populating the Dynamic Workload Console database for MSSQL



You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in FAQ - Database customizations *(on page 60)*.

You can run the configureDb command specifying a typical set of parameters. In this case, default values are used for all remaining parameters. By default, MSSQL authentication is used. To modify the authentication type, see How can I specify the authentication type when using an MSSQL database? *(on page 64)*.

For more information about all parameters and supported values of the configureDb command, see Database configuration - configureDb script *(on page 301)*. If you need to modify any of the default values, edit the `configureDbMSSQL.properties` file, but do not modify the `configureDbMSSQL.template` file located in the same path. For an example of a properties file, see What is the content of a database properties file? *(on page 66)*.

Default values are stored in the `configureDbMSSQL.properties` file, located in *`image_location`*.

Note: Only on Windows systems hosting an MSSQL database, the path hosting the tablespace must be existing before you run the configureDb.vbs command.

To create the Dynamic Workload Console database and schema, perform the following steps:

1. Only on Windows systems hosting an MSSQL database, create the path for hosting the following tablespace, if the path is not already existing:
   - TWS_DATA
2. Only on Windows systems hosting an MSSQL database, specify the path to the folder when running the configureDb.vbs command or when filling in the `configureDbMSSQL.properties` properties file with the following parameter:
   - --iwstspath
3. On the server where you plan to install the Dynamic Workload Console, extract the Dynamic Workload Console package to a directory of your choice.
4. To populate the Dynamic Workload Console database with typical settings, type the following command:

   On Windows operating systems

   ```
   cscript configureDb.vbs --rdbmstype MSSQL --dbname db_name
   --dbhostname db_hostname --dbadminuser db_administrator
   --dbadminuserpw db_administrator_password
   --iwstspath DATA_tablespace_path
   ```

   On UNIX operating systems

   ```
   ./configureDb.sh --rdbmstype MSSQL --dbname db_name
   --dbhostname db_hostname --dbadminuser db_administrator
   --dbadminuserpw db_administrator_password
   --iwstspath DATA_tablespace_path
   ```

   where:

   --rdbmstype

   The database vendor.

   --dbname *db_name*

   The name of the Dynamic Workload Console database.

   --dbhostname *db_hostname*

   The host name or IP address of database server.

--dbadminuser ***db_admin_user***

The database administrator user that creates the Dynamic Workload Console schema objects on the database server.

--dbadminuserpw ***db_admin_password***

The password of the DB administrator user that creates the IBM® Workload Scheduler schema objects on the database server. Special characters are not supported.

--iwstspath|-tp ***table_space***

The path of the tablespace for IBM® Workload Scheduler or Dynamic Workload Console data. This parameter is optional. The default value for all databases other than Oracle is:

For all operating systems, except z/OS

TWS_DATA

For z/OS operating system

TWSDATA

Only on Windows systems hosting an MSSQL database, ensure the folder for the tablespace is already existing before running the configureDb command and specify the path using this parameter. Specify the path using forward slashes (/), for example: `c:/<my_path>/TWS_DATA`.

Note: The following parameters specified with the configureDb command are also required when installing the Dynamic Workload Console and their values must be the same:

- **rdbmstype**
- **dbhostname**
- **dbport**
- **dbname**

When **--rdbmstype** is set to `MSSQL`, the default value is **sa**. To install a Dynamic Workload Console with a user different from **sa**, you must create a new user in `MSSQL` and grant all the required permissions before running the configureDb command.

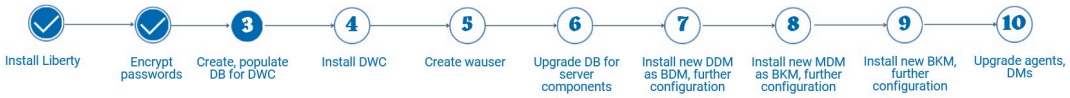
You have now successfully created and populated the Dynamic Workload Console database.

For more information about all parameters and supported values of the configureDb command, see Database configuration - configureDb script *(on page 301)*.

You can now proceed to Installing the Dynamic Workload Console *(on page 236)*.

Creating and populating the database for MSSQL cloud-based databases for the Dynamic Workload Console

Instructions for creating and populating the Dynamic Workload Console database for MSSQL cloud-based databases



MSSQL cloud-based databases include the following:

- Azure SQL
- Google Cloud SQL for SQL server
- Amazon RDS for MSSQL

You can perform a typical database procedure, as described in the following scenarios, or you can customize the database parameters, as described in FAQ - Database customizations *(on page 60)*.

You can run the configureDb command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

For more information about all parameters and supported values of the configureDb command, see Database configuration - configureDb script *(on page 301)*. If you need to modify any of the default values, edit the `configureDbMSSQL.properties` file, but do not modify the `configureDbMSSQL.template` file located in the same path. For an example of a properties file, see What is the content of a database properties file? *(on page 66)*.

Default values are stored in the `configureDbMSSQL.properties` file, located in *`image_location`*.

To create the Dynamic Workload Console database and schema, perform the following steps:

1. Specify the path to the folder when running the configureDb command or when filling in the `configureDbMSSQL.properties` properties file with the following parameter:
   - --iwstname PRIMARY

   You can optionally modify the `PRIMARY` default value when running the configureDb command.
2. On the server where you plan to install the Dynamic Workload Console, extract the Dynamic Workload Console package to a directory of your choice.
3. To populate the Dynamic Workload Console database with typical settings, type the following command:

   On Windows operating systems

   ```
   cscript configureDb.vbs --rdbmstype MSSQL --dbname db_name
   --dbhostname db_hostname --dbadminuser db_administrator
   --dbadminuserpw db_administrator_password
   --iwstsname DATA_tablespace_name
   ```

   iwstsname *DATA_tablespace_name*

   The name of the tablespace for Dynamic Workload Console data. This parameter is required.

Note: The following parameters specified with the **configureDb** command are also required when installing theDynamic Workload Console and their values must be the same:
- **rdbmstype**
- **dbhostname**
- **dbport**
- **dbname**
- **dbuser**

You have now successfully created and populated the Dynamic Workload Console database.

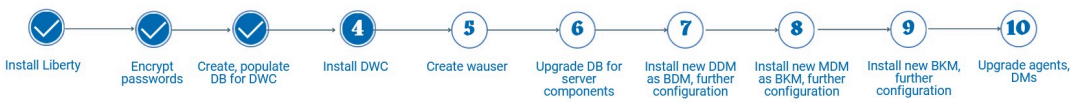
For more information about all parameters and supported values of the configureDb command, see Database configuration - configureDb script *(on page 301)*.

You can now proceed to Installing the Dynamic Workload Console *(on page 236)*.

Installing the Dynamic Workload Console

Procedure for installing two Dynamic Workload Console servers on two separate nodes.

Figure 12. Install fresh Dynamic Workload Console



The procedure to perform a fresh installation is demonstrated through a typical scenario where two Dynamic Workload Console servers are installed on separate workstations, sharing the same remote database.

With Version 9.5, the Dynamic Workload Console is based on a new architectural foundation that does not include Jazz for Service Management nor Dashboard Application Services Hub, therefore, no direct upgrade procedure is supported, but you perform a fresh installation of the Dynamic Workload Console at version 9.5.0.*x* or 10.2.*x*.

Note: If you are installing the Dynamic Workload Console version 10.2.3 or later, the Federator is also automatically installed. This component enables you to monitor your objects through the Orchestration Monitor page of the Dynamic Workload Console. For detailed information about how to configure and use the Federator, see Mirroring the z/OS current plan to enable the Orchestration Monitor *(on page)*.

If you are currently using Derby, you need to install a supported database and migrate your data. This is necessary because Derby is no longer supported as of version 10.2.3. For more information, see Connecting the Dynamic Workload Console to a new node or database *(on page 161)*.

In this scenario, the IBM® Workload Scheduler administrator installs two Dynamic Workload Console instances on two separate workstations, sharing the same remote database. The IBM® Workload Scheduler administrator performs the operations listed below on both workstations.

Certificates are now required when installing or upgrading IBM® Workload Scheduler. You can no longer install nor upgrade IBM® Workload Scheduler without securing your environment with certificates. The required certificates are:

- ca.crt
- tls.key
- tls.crt

For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).

Convert the certificates as described in Converting default certificates *(on page 223)* and copy them locally.

The IBM® Workload Scheduler administrator installs the Dynamic Workload Console. The following information is required:

Table 17. Required information

Command parameter	Required information	Provided in..
Database information		
--rdbmstype	database type	Creating and populating the database for the Dynamic Workload Console *(on page 227)*
--dbhostname	database hostname	
--dbport	database port	
--dbname	database name	
--dbuser	database user name	
--dbpassword	database password	
WebSphere Application Server Liberty Base information		
--wlpdir	WebSphere Application Server Liberty Base installation directory	Installing WebSphere Application Server Liberty *(on page 225)*
Security information		
--sslkesfolder	location of converted certificates	Converting default certificates *(on page 223)*
--sslpassword	password of converted certificates	Converting default certificates *(on page 223)*

You can run the **dwcinst** command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

Default values are stored in the `dwcinst.properties` file, located in the root directory of the installation image.

If you need to modify any of the default values, edit the `dwcinst.properties` file, but do not modify the `dwcinst.template` file located in the same path.

In a typical installation scenario, it is recommended you install the Dynamic Workload Console as a **non-root user** on UNIX systems and as a **local administrator** on Windows systems.

This user is automatically created by the installation process in the WebSphere Application Server Liberty Base repository. Ensure that the user has full access to the WebSphere Application Server Liberty Base installation directory.

To install the Dynamic Workload Console, perform the following steps:

1. Log in to the workstation where you plan to install the Dynamic Workload Console.
2. Download the installation images from IBM Fix Central.
3. Browse to the folder where the `dwcinst` command is located in *`image_location`*`/TWS/`*`interp_name`*.
4. Start the installation specifying a typical set of parameters:

   On Windows operating systems

   ```
   cscript dwcinst.vbs --acceptlicense yes --rdbmstype db_type
   --user dwc_admin_user --password dwc_pwd --dbname db_name
   --dbuser db_user --dbpassword db_pwd --dbhostname db_hostname
   --dbport db_port --wlpdir Liberty_installation_dir\wlp
   --sslkeysfolder certificate_files_path --sslpassword keystore_truststore_password
   ```

   On UNIX operating systems

   ```
   ./dwcinst.sh --acceptlicense yes --rdbmstype db_type
   --user dwc_admin_user --password dwc_pwd --dbname db_name
   --dbuser db_user --dbpassword db_pwd --dbhostname db_hostname
   --dbport db_port --wlpdir Liberty_installation_dir/wlp
   --sslkeysfolder certificate_files_path --sslpassword keystore_truststore_password
   ```

   where,

   user *dwc_admin_user*

   is the administrator of the Dynamic Workload Console. This user is added to the group of the Dynamic Workload Console administrators at installation time. You can use this account to log in to the Dynamic Workload Console and manage your environment.

   password *dwc_pwd*

   is the password of the Dynamic Workload Console user.

   On Windows operating systems

   Supported characters for the password are alphanumeric, dash (-), underscore (_) characters, and ()|?*~+.@!^

   On UNIX operating systems

   Supported characters for the password are any alphanumeric, dash (-), underscore (_) characters, and ()|?=*~+.

You have now successfully installed the Dynamic Workload Console.

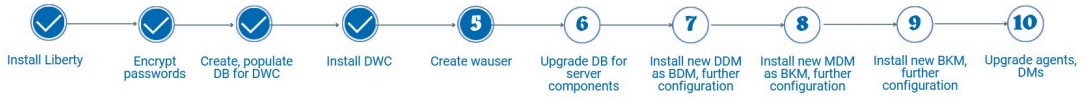
For more information about all **dwcinst** parameters and default values, see Dynamic Workload Console installation - dwcinst script *(on page 320)*.

If you had previously exported the Dynamic Workload Console, as described in Connecting the Dynamic Workload Console to a new node or database *(on page 161)*, you can now import them in the new Dynamic Workload Console from the **Administration > Manage Settings** menu. If you have a high availability configuration, import the settings on one node.

You can now proceed to Creating the IBM Workload Scheduler administrative user *(on page 239)*.

Creating the IBM® Workload Scheduler administrative user

Instructions to create the IBM® Workload Scheduler administrative user



IBM® Workload Scheduler administrative user

The IBM® Workload Scheduler administrator creates the administrative user (**wauser**). The administrative user is the user for which the product will be installed in the subsequent steps. This implies that this user has full access to all scheduling objects.

The user name can contain alphanumeric, dash (-), and underscore (_) characters; it cannot contain national characters. The first character of the user name must be a letter.

The following considerations apply:

On Windows operating systems:

- If this user account does not already exist, it is automatically created at installation time.
- If installing on a Windows™ server in a domain, do not define a domain and local ID with the same user name.
- If you specify a domain user, define the name as *domain_name**user_name*.
- If you specify a local user, define the name as *system_name**user_name*. Type and confirm the password.

On UNIX and Linux operating systems:

This user account must be created manually before running the installation and must be enabled to login to the machine where the master domain manager is going to be installed. Create a user with a home directory and group. Use the appropriate UNIX and Linux operating system commands to create the user.

Important: Group names that contain a "/" (forward slash) character can cause permissions to not be set correctly. When IBM® Workload Scheduler retrieves credentials from WebSphere Application Server Liberty, it parses the returned list of groups names assuming they are saved in the format `<realm_name>/<group_name>`. If the group name, the realm name, or both contain a "/" character, the parsing fails.

You can also install IBM® Workload Scheduler using a user different from the root user. This installation method is known as **no-root installation** and applies to all IBM® Workload Scheduler components. Note that if you choose this installation method, only the user who performs the installation can use IBM® Workload Scheduler. For this reason, the typical installation scenario described in this section uses the root user.

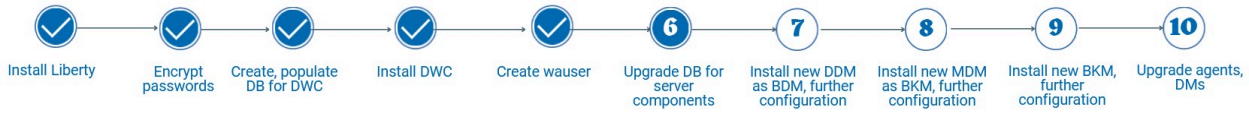
For more information, see IBM Workload Scheduler user management *(on page 34)*.

What to do next

You can now proceed to Upgrading the database for the server components *(on page 239)*.

Upgrading the database for the server components

Upgrade the master domain manager database tables before upgrading the server



components.

Note: Before upgrading the database schema, ensure you have created a backup. Refer to the documentation related to your RDBMS for information about the backup procedure.

Ensure you have acquired information about the IBM® Workload Scheduler tablespaces that were specified when the database tables were created and populated the first time. If values different from the default values were used, then your database administrator must provide them for this upgrade procedure. If default values were used, then they do not need to be specified during the upgrade procedure. The default values for the IBM® Workload Scheduler data, log, and plan tablespaces are as follows:

- **--iwstsname** `TWS_DATA`
  - For Oracle only, the default is `USERS`
- **--iwslogtsname** `TWS_LOG`
  - For Oracle only, the default is `USERS`
- **--iwsplantsname** `TWS_PLAN`
  - For Oracle only, the default is `USERS`

For more information about all parameters and supported values of the `configureDb` command, see Database configuration - configureDb script *(on page 301)*.

You can run the configureDb command specifying a typical set of parameters. In this case, default values are used for all remaining parameters.

The script creates an SQL file with all the statements needed to upgrade the IBM® Workload Scheduler database schema to the latest version and, by default, automatically applies it.

Default values are stored in the `configureDb<`*`database_vendor`*`>.properties` file, located in *`image_location`*`/TWS/`*`interp_name`*. For an example of a properties file, see What is the content of a database properties file? *(on page 66)*.

If you need to modify any of the default values, edit the `configureDb<`*`database_vendor`*`>.properties` file, but do not modify the `configureDb<`*`database_vendor`*`>.template` file located in the same path.

To upgrade the IBM® Workload Scheduler database schema, perform the following steps:

1. On the workstation where you plan to install the new backup master domain manager or backup dynamic domain manager, extract the IBM® Workload Scheduler package at the latest version to a directory of your choice.
2. Browse to the *`image_location`*`/TWS/`*`interp_name`* path.
3. Type the following command to upgrade the IBM® Workload Scheduler database schema to the latest version. Ensure that you use the same database administrator credentials you used when the IBM® Workload Scheduler database schema objects were created. The new backup master domain manager or backup dynamic domain manager is configured to point to the existing database instance.

   On Windows operating systems

   ```
   cscript configureDb.vbs --rdbmstype db_vendor --dbhostname db_hostname --dbport db_port
   --dbname db_name --dbuser db_user  --componenttype server_component
   --dbadminuser db_administrator --dbadminuserpw db_administrator_password
   --iwstsname tablespace_data --iwslogtsname tablespace_log --iwsplantsname tablespace_plan
   ```

   On UNIX operating systems

   ```
   ./configureDb.sh --rdbmstype db_vendor --dbhostname db_hostname --dbport db_port
   --dbname db_name --dbuser db_user  --componenttype server_component
   --dbadminuser db_administrator --dbadminuserpw db_administrator_password
   --iwstsname tablespace_data --iwslogtsname tablespace_log --iwsplantsname tablespace_plan
   ```

   where:

   --rdbmstype

   The database vendor.

   --dbhostname *db_hostname*

   The host name or IP address of database server.

--dbport ***db_port***

The port of the database server.

--dbname ***db_name***

The name of the IBM® Workload Scheduler database.

--dbuser ***db_user***

The user that has been granted access to the IBM® Workload Scheduler tables on the database server.

--dbpassword ***db_password***

The password for the user that has been granted access to the IBM® Workload Scheduler tables on the database server. Special characters are not supported.

--dbadminuser ***db_admin_user***

The database administrator user that creates the IBM® Workload Scheduler schema objects on the database server.

--dbadminuserpw ***db_admin_password***

The password of the DB administrator user that creates the IBM® Workload Scheduler schema objects on the database server. Special characters are not supported.

--componenttype MDM | DDM

The IBM® Workload Scheduler component for which the database is installed. This parameter is optional. Supported values are:

MDM

master domain manager.

DDM

dynamic domain manager.

--iwstsname ***tablespace_data***

The name of the tablespace for IBM® Workload Scheduler data. The default value for all supported RDBMS is TWS_DATA, with the exception of Oracle where the default is USERS.

--iwslogtsname ***tablespace_log***

The name of the tablespace for the IBM® Workload Scheduler log. The default value for all supported RDBMS is TWS_LOG, with the exception of Oracle where the default is USERS.

--iwsplantsname ***db_port***

The name of the tablespace for the IBM® Workload Scheduler plan. The default value for all supported RDBMS is TWS_PLAN, with the exception of Oracle where the default is USERS.

--auth_type ***db_name***

The MSSQL authentication mode. The default is SQLSERVER which uses native SQL authentication.

You can optionally point the backup master domain manager to different database residing on the sane workstation. For more information, see Connecting the master domain manager to a new database *(on page)*.

Note: The following parameters specified with the configureDb command are also required when you upgrade the server components with the serverinst command and their values must be the same:

- **rdbmstype**
- **dbhostname**
- **dbport**
- **dbname**
- **dbuser**
- **dbpassword**

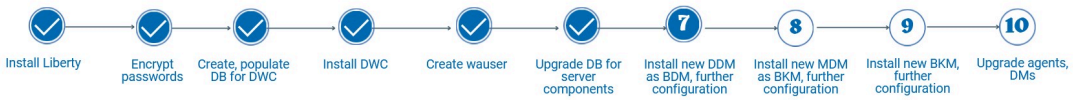
You have now successfully upgraded the database schema for the IBM® Workload Scheduler database.

You can now proceed to Installing a new dynamic domain manager configured as a backup *(on page 242)* or to Installing the new master domain manager configured as a backup *(on page 247)*.

Installing a new dynamic domain manager configured as a backup

Install a new dynamic domain manager configured as a backup and link it to your current network. Then switch it to become the new dynamic domain manager.

This is a parallel upgrade procedure that installs a fresh dynamic domain manager configured as backup. The dynamic domain manager configured as a backup points to your existing IBM® Workload Scheduler database and then later becomes your new dynamic domain manager.



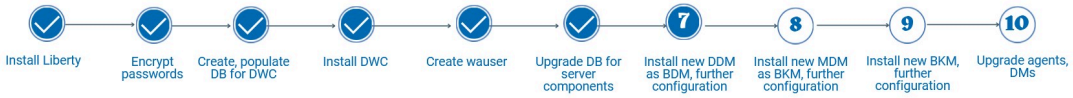
This section describes how to upgrade the dynamic components (dynamic domain manager and its backup). For details about the supported versions from which you can upgrade, see the IBM Workload Scheduler Release Notes.

Perform the following steps:

1. Installing a new dynamic domain manager configured as a backup *(on page 242)*
2. Switching the dynamic domain manager to the new dynamic domain manager configured as backup *(on page 244)*
3. Installing a new backup dynamic domain manager *(on page 245)* to replace the backup dynamic domain manager which you have switched to become the current dynamic domain manager.
4. Switching back to the old dynamic domain manager (optional) *(on page 247)*

Installing a new dynamic domain manager configured as a backup

Procedure for installing a dynamic domain manager configured as a backup



Install a new dynamic domain manager at the latest product version level configured as the new backup dynamic domain manager by running the serverinst script.

The procedure to install the dynamic domain manager and backup dynamic domain manager is exactly the same, with the difference that it is performed on two different workstations and that each installation points to its local WebSphere Application Server Liberty Base installation. IBM® Workload Scheduler determines whether or not a dynamic domain manager is already present in the environment and proceeds to install a dynamic domain manager or backup dynamic domain manager accordingly.

The IBM® Workload Scheduler administrator installs the dynamic domain manager as the backup. The following information is required:

Table 18. Required information

Required information for performing the installation

Command parameter	Information type	Provided in...
Database information		
--rdbmstype	database type	Upgrading the database for the server components *(on page 239)*
--dbhostname	database hostname	

Table 18. Required information

Required information for performing the installation

(continued)

--dbport	database port	
--dbname	database name	
--dbuser	database user name	
--dbpassword	database password	
IBM® Workload Scheduler information		
--wauser	IBM® Workload Scheduler administrative user name	Creating the IBM Workload Scheduler administrative user *(on page 239)*
--wapassword	IBM® Workload Scheduler administrative user password	
WebSphere Application Server Liberty Base information		
--wlpdir	WebSphere Application Server Liberty Base installation directory	Installing WebSphere Application Server Liberty *(on page 225)*
Security information		
--sslkeysfolder	location of converted certificates	Converting default certificates *(on page 223)*
--sslpassword	password of converted certificates	Converting default certificates *(on page 223)*

Before starting the installation, ensure the following steps have been completed:

1. Converting default certificates *(on page 223)*. Because you are installing a dynamic domain manager, also copy locally the `jwt.crt` file created in the conversion procedure.
2. Installing WebSphere Application Server Liberty *(on page 225)* on the workstation where you plan to install the dynamic domain manager and on the workstation where you plan to install the backup dynamic domain manager.
3. Encrypting passwords (optional) *(on page 226)*
4. Upgrading the database for the server components *(on page 239)*
5. Creating the IBM Workload Scheduler administrative user *(on page 239)*

You can run the **serverinst** command specifying a typical set of parameters. In this case, default values are used for all remaining parameters. For more information about all **serverinst** parameters and default values, see Server components installation - serverinst script *(on page 310)*.

A properties file named `serverinst.properties` is available if you do not want to type parameters in the command line. This is expecially useful if you need to specify many parameters or if you want to reuse the file for several installations. The file is located in *`image_location`*`/TWS/`*`interp_name`*.

You can specify values in the properties file, type them in the command line, or use both methods. If a parameter is specified both in the properties file and in the command line, the command line value takes precedence.

If you need to modify any of the default values, edit the `serverinst.properties` file, but do not modify the `serverinst.template` file located in the same path.

To install the dynamic domain manager as a backup, perform the following steps:

1. Log in to the workstation where you plan to install.
2. Browse to the folder where the `serverinst` command is located:

   On Windows operating systems

image_location\TWS*interp_name*

On UNIX operating systems

image_location/TWS/*interp_name*

3. Start the installation specifying a typical set of parameters. In this case, default values are used for all remaining parameters:

   On Windows operating systems

   ```
   cscript serverinst.vbs --acceptlicense yes --rdbmstype db_type
           --dbhostname db_hostname --dbport db_port --dbname db_name
           --dbuser db_user --dbpassword db_password --wauser wa_user
           --wapassword wa_password --componenttype DDM --domain domain_name
           --master mdm_name --mdmbrokerhostname mdm_broker_host_name
           --sslkeysfolder certificate_files_path --sslpassword keystore_truststore_password
           --mdmhttpsport mdm_https_host_name --wlpdir Liberty_installation_dir\wlp
   ```

   On UNIX operating systems

   ```
   ./serverinst.sh --acceptlicense yes --rdbmstype db_type
           --dbhostname db_hostname --dbport db_port --dbname db_name
           --dbuser db_user --dbpassword db_password --wauser wa_user
           --wapassword wa_password --componenttype DDM --domain domain_name
           --master mdm_name --mdmbrokerhostname mdm_broker_host_name
           --sslkeysfolder certificate_files_path --sslpassword keystore_truststore_password
           --mdmhttpsport mdm_https_host_name --wlpdir Liberty_installation_dir/wlp
   ```

4. Distribute the Symphony file to the new dynamic domain manager configured as backup:
   a. Ensure that the **optman cf** option is set to *all*.
   b. To distribute the Symphony file to the new dynamic domain manager configured as backup, run JnextPlan -for 0000 or wait until the end of the production plan.
   c. Restore the previous setting of the **optman cf** option, if you previously modified the value.

You have now successfully installed the backup dynamic domain manager at the new product version level.

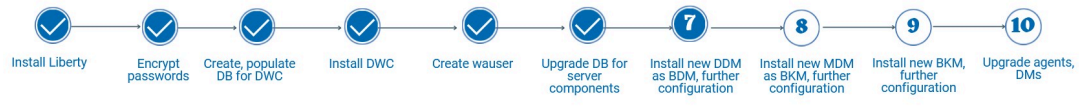
For more information about all **serverinst** parameters and default values, see Server components installation - serverinst script *(on page 310)*.

What to do next

You can now proceed to Switching the dynamic domain manager to the new dynamic domain manager configured as backup *(on page 244)*.

Switching the dynamic domain manager to the new dynamic domain manager configured as backup

Switch the old dynamic domain manager to become a backup dynamic domain manager. As a result, the backup dynamic domain manager you installed in the previous step, becomes the current dynamic domain manager.



Switch to your new dynamic domain manager configured as backup, so that it becomes your current dynamic domain manager, by completing the following steps:

1. Stop the workload broker server on the dynamic domain manager at the previous product version level, by running the following command:

   On Windows operating systems

```
stopBrokerApplication.bat
 -user username -password password
[-port portnumber]
```

On UNIX and Linux operating systems

```
stopBrokerApplication.sh
 -user username -password password
[-port portnumber]
```

where *username* and *password* are the values specified during the dynamic domain manager installation. The parameter *portnumber* is optional, if it is not specified, the default is used.

2. Switch the dynamic domain manager to its backup workstation. Use either the Dynamic Workload Console or run the command:

```
conman
switchmgr dyn_dom;new_mgr_cpu
```

where *dyn_dom* is the domain where you installed the backup dynamic domain manager and the *new_mgr_cpu* is the backup dynamic domain manager workstation name.

3. From the new current dynamic domain manager, unlink the old dynamic domain manager workstation:

```
conman "unlink old_ddm_wks"
```

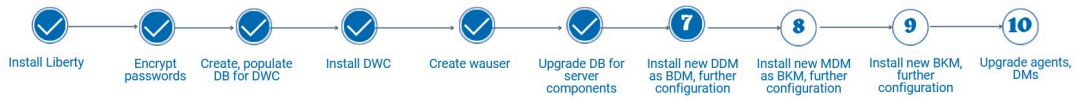
where *old_ddm_wks* is the old dynamic domain manager workstation name at the previous product version that now has the backup role.

For more detailed information about switching a domain manager, see Complete procedure for switching a domain manager *(on page)*.

You can now proceed to install a new dynamic domain manager configured as a backup at the latest production version. as described in Installing a new backup dynamic domain manager *(on page 245)*

Installing a new backup dynamic domain manager

Procedure for installing the new backup dynamic domain manager



At this phase in the procedure, you have installed a fresh backup dynamic domain manager at the latest product version level and switched it to become the new dynamic domain manager. To complete the environment set up, you now need to install a new backup dynamic domain manager at the latest product version level by running the serverinst script.

The procedure to install the dynamic domain manager and backup dynamic domain manager is exactly the same, with the difference that it is performed on two different workstations and that each installation points to its local WebSphere Application Server Liberty Base installation. IBM® Workload Scheduler determines whether or not a dynamic domain manager is already present in the environment and proceeds to install a dynamic domain manager or backup dynamic domain manager accordingly.

The IBM® Workload Scheduler administrator installs the dynamic domain manager as the backup. The following information is required:

Table 19. Required information

Required information for performing the installation

Command parameter	Information type	Provided in...
	IBM® Workload Scheduler information	

Table 19. Required information

Required information for performing the installation

(continued)

--wauser	IBM® Workload Scheduler administrative user name	Creating the IBM Workload Scheduler administrative user *(on page 239)*
--wapassword	IBM® Workload Scheduler administrative user password	
WebSphere Application Server Liberty Base information		
--wlpdir	WebSphere Application Server Liberty Base installation directory	Installing WebSphere Application Server Liberty *(on page 225)*
Security information		
--sslkeysfolder	location of converted certificates	Converting default certificates *(on page 223)*
--sslpassword	password of converted certificates	Converting default certificates *(on page 223)*

Before starting the installation, ensure the following steps have been completed:

1. Converting default certificates *(on page 223)*. Because you are installing a dynamic domain manager, also copy locally the `jwt.crt` file created in the conversion procedure.
2. Installing WebSphere Application Server Liberty *(on page 225)* on the workstation where you plan to install the dynamic domain manager and on the workstation where you plan to install the backup dynamic domain manager.
3. Encrypting passwords (optional) *(on page 226)*
4. Upgrading the database for the server components *(on page 239)*
5. Creating the IBM Workload Scheduler administrative user *(on page 239)*

You can run the **serverinst** command specifying a typical set of parameters. In this case, default values are used for all remaining parameters. For more information about all **serverinst** parameters and default values, see Server components installation - serverinst script *(on page 310)*.

A properties file named `serverinst.properties` is available if you do not want to type parameters in the command line. This is expecially useful if you need to specify many parameters or if you want to reuse the file for several installations. The file is located in *`image_location`*`/TWS/`*`interp_name`*.

You can specify values in the properties file, type them in the command line, or use both methods. If a parameter is specified both in the properties file and in the command line, the command line value takes precedence.

If you need to modify any of the default values, edit the `serverinst.properties` file, but do not modify the `serverinst.template` file located in the same path.

To install the dynamic domain manager as a backup, perform the following steps:

1. Log in to the workstation where you plan to install.
2. Browse to the folder where the `serverinst` command is located:

   On Windows operating systems

   `image_location``\TWS\`*`interp_name`*

   On UNIX operating systems

   `image_location``/TWS/`*`interp_name`*
3. Start the installation specifying a typical set of parameters. In this case, default values are used for all remaining parameters:

   On Windows operating systems

```
cscript serverinst.vbs --acceptlicense yes --rdbmstype db_type
        --dbhostname db_hostname --dbport db_port --dbname db_name
        --dbuser db_user --dbpassword db_password --wauser wa_user
        --wapassword wa_password --componenttype DDM --domain domain_name
        --master mdm_name --mdmbrokerhostname mdm_broker_host_name
        --sslkeysfolder certificate_files_path --sslpassword keystore_truststore_password
        --mdmhttpsport mdm_https_host_name --wlpdir Liberty_installation_dir\wlp
```

On UNIX operating systems

```
./serverinst.sh --acceptlicense yes --rdbmstype db_type
        --dbhostname db_hostname --dbport db_port --dbname db_name
        --dbuser db_user --dbpassword db_password --wauser wa_user
        --wapassword wa_password --componenttype DDM --domain domain_name
        --master mdm_name --mdmbrokerhostname mdm_broker_host_name
        --sslkeysfolder certificate_files_path --sslpassword keystore_truststore_password
        --mdmhttpsport mdm_https_host_name --wlpdir Liberty_installation_dir/wlp
```

4. Distribute the Symphony file to the new dynamic domain manager configured as backup:
   a. Ensure that the **optman cf** option is set to *all*.
   b. To distribute the Symphony file to the new dynamic domain manager configured as backup, run JnextPlan -for 0000 or wait until the end of the production plan.
   c. Restore the previous setting of the **optman cf** option, if you previously modified the value.

You have now successfully installed the backup dynamic domain manager at the new product version level.

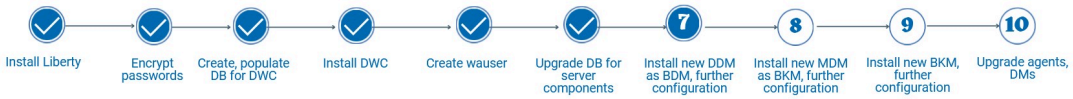
For more information about all **serverinst** parameters and default values, see Server components installation - serverinst script *(on page 310)*.

What to do next

You can now optionally proceed to Switching back to the old dynamic domain manager (optional) *(on page 247)*.

Switching back to the old dynamic domain manager (optional)

Optionally switch back to the old dynamic domain manager



This step is optional. You can switch back to your old dynamic domain manager.

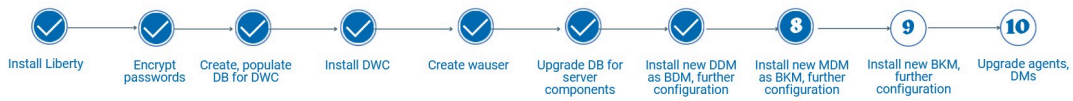
From the old dynamic domain manager, run the command:

```
conman
switchmgr dyn_dom;old_mgr_cpu
```

where *dyn_dom* is the domain where the dynamic domain manager configured as backup is installed and the *old_mgr_cpu* is the old dynamic domain manager workstation name

Installing the new master domain manager configured as a backup

Install the new master domain manager configured as a backup and link it to your current network. Then switch it to become the new master domain manager.



Complete the steps listed below to install a fresh master domain manager configured as backup and then link it to your current network.

The master domain manager configured as a backup points to your existing IBM Workload Scheduler database and then later becomes your new master domain manager.

During the master domain manager upgrade process, the license model to be applied to the environment is defined. The license model determines the criteria by which your license compliance is calculated. The following pricing models are supported: **byWorkstation**, **perServer**, **perJob**. The default value is **perServer**. To determine the current value of this global option, enter the following command: **optman show ln** or **optman show licenseType**. To modify the pricing model, use the **optman chg ln** or **optman chg licenseType** command. For more information about licensing, see License Management in IBM License Metric Tool *(on page)* .

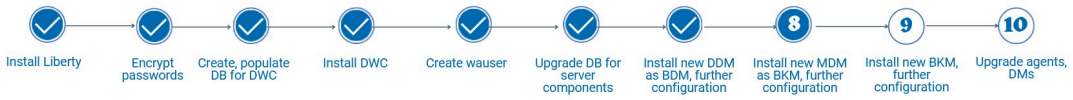
1. Converting default certificates *(on page 223)*, if you are using default certificates in your current environment. Use this procedure to convert the certificates from the JKS to the PEM format, then copy them to the workstations where you plan to install the server components (dynamic domain manager and its backups, master domain manager and its backups) and the Dynamic Workload Console.

   Certificates are now required when installing or upgrading IBM® Workload Scheduler. You can no longer install nor upgrade IBM® Workload Scheduler without securing your environment with certificates. The required certificates are:
   - ca.crt
   - tls.key
   - tls.crt

   For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).
2. Installing the master domain manager as a backup master domain manager *(on page 248)*
3. Switching the master domain manager to the new backup master *(on page 252)*
4. Making the switch manager permanent *(on page 253)*
5. Customizing and submitting the optional FINAL job stream *(on page 254)*

Installing the master domain manager as a backup master domain manager

A fresh installation for the master domain manager and the backup master domain manager



Before beginning the installation, ensure you have completed the following steps:

1. Converting default certificates *(on page 223)*
2. Installing WebSphere Application Server Liberty *(on page 225)*
3. Encrypting passwords (optional) *(on page 226)*
4. Upgrading the database for the server components *(on page 239)*
5. Creating the IBM Workload Scheduler administrative user *(on page 239)*

You install a master domain manager at the latest product version level configured as the new backup master domain manager by running the serverinst script. The installation process is able to detect the presence of an existing master domain manager and automatically configures this one as the backup master domain manager. The new backup master domain manager is configured to point to the existing database instance.

The IBM® Workload Scheduler administrator installs the master domain manager as the backup. The following information is required:

Table 20. Required information

Required information for performing the installation

Command parameter	Information type	Provided in..
Database information		
--rdbmstype	database type	Upgrading the database for the server components *(on page 239)*
--dbhostname	database hostname	
--dbport	database port	
--dbname	database name	
--dbuser	database user name	
--dbpassword	database password	
IBM® Workload Scheduler information		
--wauser	IBM® Workload Scheduler administrative user name	Creating the IBM Workload Scheduler administrative user *(on page 239)*
--wapassword	IBM® Workload Scheduler administrative user password	
WebSphere Application Server Liberty Base information		
--wlpdir	WebSphere Application Server Liberty Base installation directory	Installing WebSphere Application Server Liberty *(on page 225)*
IBM® Workload Scheduler installation directory		
--inst_dir	installation directory	Current procedure
Security information		
--sslkeysfolder	location of converted certificates	Converting default certificates *(on page 223)*
--sslpassword	password of converted certificates	Converting default certificates *(on page 223)*

You can run the **serverinst** command specifying a typical set of parameters. In this case, default values are used for all remaining parameters. For more information about all **serverinst** parameters and default values, see Server components installation - serverinst script *(on page 310)*.

A properties file named `serverinst.properties` is available if you do not want to type parameters in the command line. This is expecially useful if you need to specify many parameters or if you want to reuse the file for several installations. The file is located in *`image_location`*`/TWS/`*`interp_name`*.

You can specify values in the properties file, type them in the command line, or use both methods. If a parameter is specified both in the properties file and in the command line, the command line value takes precedence.

If you need to modify any of the default values, edit the `serverinst.properties` file, but do not modify the `serverinst.template` file located in the same path.

To install the master domain manager as a backup, perform the following steps:

1. Log in to the workstation where you plan to install the master domain manager.
2. Browse to the folder where the `serverinst` command is located in *`image_location`*`/TWS/`*`interp_name`*.
3. Start the installation specifying a typical set of parameters. In this case, default values are used for all remaining parameters:

On Windows operating systems

```
cscript serverinst.vbs --acceptlicense yes --rdbmstype <db_type>
--dbhostname <db_hostname> --dbport <db_port> --dbname <db_name>
--dbuser <db_user> --dbpassword <db_password> --wauser <wa_user>
--wapassword <wa_password> --wlpdir <Liberty_installation_dir>\wlp
--sslkeysfolder <certificate_files_path> --sslpassword <keystore_truststore_password>
--inst_dir <installation_dir>
```

On UNIX operating systems

```
./serverinst.sh --acceptlicense yes --rdbmstype <db_type>
--dbhostname <db_hostname> --dbport <db_port> --dbname <db_name>
--dbuser <db_user> --dbpassword <db_password> --wauser <wa_user>
--wapassword <wa_password> --wlpdir <Liberty_installation_dir>/wlp
--sslkeysfolder <certificate_files_path> --sslpassword <keystore_truststore_password>
--inst_dir <installation_dir>
```

where

--acceptlicense

Specify **yes** to accept the product license.

--rdbmstype|-r *rdbms_type*

The database type. Supported databases are:

- **DB2**
- **ORACLE** This value applies to Oracle and Amazon RDS for Oracle
- **MSSQL** This value applies to MSSQL and MSSQL cloud-based databases.
- **POSTGRESQL**

This parameter is required and has no default value.

--dbhostname *db_hostname*

The host name or IP address of database server.

--dbport *db_port*

The port of the database server.

--dbname *db_name*

The name of the IBM® Workload Scheduler database.

--dbuser *db_user*

The database user that has been granted access to the IBM® Workload Scheduler tables on the database server.

--dbpassword *db_password*

The password for the user that has been granted access to the IBM® Workload Scheduler tables on the database server. Special characters are not supported.

--wauser *user_name*

The user for which you are installing IBM Workload Scheduler.

--wapassword *wauser_password*

The password of the user for which you are installing IBM Workload Scheduler.

On Windows operating systems

Supported characters for the password are alphanumeric, dash (-), underscore (_) characters, and ()|?*~+.@!^

On UNIX operating systems

Supported characters for the password are any alphanumeric, dash (-), underscore (_) characters, and ()|?=*~+.

--wlpdir

The path where WebSphere Application Server Liberty Base is installed.

--sslkeysfolder ***keystore_truststore_folder***

The name and path of the folder containing certificates in PEM format. The installation program automatically processes the keystore and truststore files using the password you specify with the **--sslpassword** parameter. The folder must contain the following files:

- **ca.crt**

  The Certificate Authority (CA) public certificate. Note that if certificates being installed are part of a chain consisting of 3 or more certificates (one Root CA, followed by one or more Intermediate CAs, followed by the end user certificate), then this file must contain the Root CA certificate only. Any Intermediate CA certificates must be stored in the `additionalCAs` subfolder, which therefore becomes a mandatory subfolder. Each Intermediate CA must be stored in the `additionalCAs` subfolder in its own file.

  Note: From V10.2.3, if certificates being installed are part of a chain, the ca.crt can contain also the intermediate CAs. In this case, it must begin with one or more intermediate CA certificates and end with the Root ca.

- **tls.key**

  The private key of the end user certificate for the instance to be installed.

- **tls.crt**

  The public part of the previous key, that is the end user certificate.

For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).

You can optionally create a subfolder to contain one or more `*.crt` files to be added to the server truststore as trusted CA, whose name must be `additionalCAs`. This can be used for example to add to the list of trusted CAs the certificate of the LDAP server or DB2 server. Additionally, you can store here any intermediate CA certificate to be added to the truststore. The subfolder must be named **additionalCAs**. Note that if the end user certificate being installed in the instance is part of a chain consisting of 3 or more certificates (one Root CA, followed by one or more Intermediate CAs, followed by the end user certificate), then the Intermediate CAs certificates must be stored in the `additionalCAs` subfolder, which therefore becomes a mandatory subfolder. Each Intermediate CA must be stored in the `additionalCAs` subfolder in its own file.

For further information about how to generate custom certificates, see Managing certificates using Certman *(on page)*.

--sslpassword ***ssl_password***

The password for the custom certificates and the path to the folder containing certificates in PEM format with the **sslkeysfolder** parameter.

For more information, see sslkeysfolder *(on page 316)*.

You can optionally encrypt the password using the secure script. For more information, see Optional password encryption - secure script *(on page 300)*.

--inst_dir ***installation_dir***

The directory of the IBM Workload Scheduler installation.

Note: The values for the following parameters must match the values you provided when creating and populating the database:

- **--rdbmstype**
- **--dbhostname**
- **--dbport**
- **--dbname**

- **--dbuser**
- **--dbpassword**

Note: Before starting the deployment of a new master domain manager or backup master domain manager on an already used database, be sure that no failed plan creation/extension has been performed. If a failed plan creation or extension has been performed, resolve the failure before attempting the new deployment or unlock the database by running the planman unlock db command.

4. If you are installing a backup master domain manager, it is crucial to use the same encryption keys as those on the master domain manager, to ensure it can correctly decrypt encrypted files, such as the Symphony file. To achieve this, perform the following steps:
   a. Backup the files located in the *TWA_DATA_DIR*`\ssl\aes` folder on the backup master domain manager.
   b. Copy the files from the *TWA_DATA_DIR*`\ssl\aes` folder on the master domain manager to the *TWA_DATA_DIR*`\ssl\aes` folder on the backup master domain manager.
5. To verify that the installation completed successfully, browse to the directory where you installed the master domain manager and type the following commands:

   On UNIX operating systems

   ```
   . ./tws_env.sh
   ```

   On Windows operating systems

   ```
   tws_env.cmd
   ```

```
optman ls
```

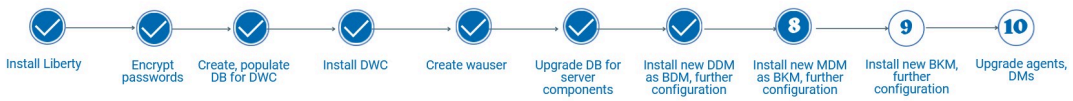
This command lists the IBM® Workload Scheduler configurations settings and confirms that IBM® Workload Scheduler installed correctly.

You can also optionally run JnextPlan -for 0000 to extend by 0 hours and 0 minutes the production plan and add into the production plan (Symphony) the newly-created workstation, or wait for the FINAL job stream to complete, then run composer list cpu=*server_workstation_name* to ensure the agents have registered. You can also run a test job to ensure everything is working correctly.

You have now successfully installed the master domain manager as the backup master domain manager.

You can now proceed to Switching the master domain manager to the new backup master *(on page 252)*.

Switching the master domain manager to the new backup master



To switch the back-level master domain manager to the new backup master domain manager, complete the following procedure:

1. Start WebSphere Application Server Liberty Base on the new backup master domain manager by running the startAppServer script found in the following path:

   ```
   <TWA_HOME>/appservertools/startAppServer.sh
   ```

2. Before you switch your master domain manager to the new backup master domain manager, you must stop the dynamic workload broker server on the current back-level master domain manager:

   On Windows™ operating systems

   Use wastool `stopBrokerApplication.bat`

   On UNIX® operating systems

Use wastool `stopBrokerApplication.sh`

3. Switch to your new backup master domain manager, which now becomes your current active master domain manager, by issuing the following command from either the Dynamic Workload Console or the **command line** of your old master domain manager:

   From the Dynamic Workload Console

   In the navigation tree, click **Monitoring and Reporting** > **Monitor Workload** > select the engine and the object type Workstation, click run and, in the table of results, select backup master domain manager workstation name, click **More Actions**, and select **Become Master Domain Manager**.

   From the command line of the old master domain manager

   Issue the following command:

   ```
   conman "switchmgr masterdm;new_mgr_cpu"
   ```

   where *new_mgr_cpu* is the backup master domain manager workstation name.

4. Switch the event processor from the old master domain manager to the backup master domain manager, by running the following command from either the Dynamic Workload Console or the **command line** of your old master domain manager:

   From the Dynamic Workload Console

   In the navigation tree, click **Monitoring and Reporting** > **Monitor Workload** > select the engine and the object type Workstation, click run and, in the table of results, select backup master domain manager workstation name, click **More Actions**, and select **Become Event Processor**.

   From the command line of the old master domain manager

   Issue the following command:

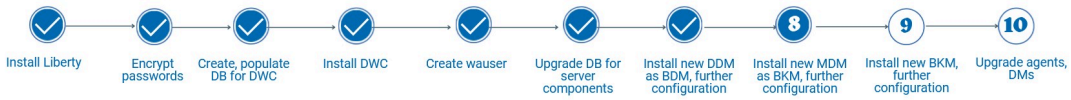
   ```
   conman "switcheventprocessor new_mgr_cpu"
   ```

   where *new_mgr_cpu* is the backup master domain manager workstation name.

Once you have switched the master domain manager to the new backup master, you can make this switch permanent. For details, see Making the switch manager permanent *(on page 253)*.

For more detailed information about switching the master domain manager, see Short-term switch of a master domain manager *(on page)*

Making the switch manager permanent



In the procedure Switching the master domain manager to the new backup master *(on page 252)*, you switched your master domain managermaster domain manager promoting your new version backup master domain manager to the role of master domain manager.

To make this configuration fully operational and persistent through **JnextPlan**, you must complete the following procedure:

On the new master domain manager, referred to as *new_mgr_cpu*, perform the following steps:

1. Edit the *localopts* file and modify the following entry as shown:

   ```
   DEFAULTWS=new_mgr_cpu
   ```

   where *new_mgr_cpu* is the workstation name of the new master domain manager. For more information about `localopts` file, see Setting local options *(on page)*.
2. Change the workstation definition of the back-level master by running:

```
composer modify cpu=old_mgr_cpu
```

and in the definition substitute *type=manager* with *type=fta*

3. Change the workstation definition of the new master by running:

```
composer modify cpu=new_mgr_cpu
```

and in the definition substitute *type=fta* with *type=manager*.

4. Ensure that the **optman** `cf` option is set to *all*.
5. Rebuild the plan to activate the changes to the database:

```
JnextPlan -for 0000
```

6. Switch the event processor to the new master domain manager by running the following command:

```
switcheventprocessor new_mgr_cpu
```

7. Restore the previous setting of the **optman** `cf` option, if necessary.
8. Edit the *TWA_DATA_DIR*`/mozart/globalopts` file and modify the **master=*old_mgr_cpu*** entry as shown:

```
master=new_mgr_cpu
```

where *new_mgr_cpu* is the workstation name of the new master. For more information about optman, see Setting global options *(on page)*.

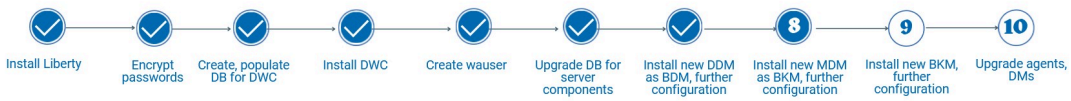
In this way the reports reptr-pre and reptr-post can run when you run **JnextPlan**.

Once you have made the switch manager permanent, you must run the FINAL job stream on the new master domain manager.

You can now proceed to Customizing and submitting the optional FINAL job stream *(on page 254)*.

Customizing and submitting the optional FINAL job stream

Merge the functions of your current FINAL and FINALPOSTREPORTS job streams with the syntax of your new FINAL and FINALPOSTREPORTS job streams.



The upgrade process writes the latest FINAL and FINALPOSTREPORTS definitions for the current release in the following file: `<`*`TWA_HOME`*`>/TWS/config/Sfinal`, where `<`*`TWA_HOME`*`>` is the IBM Workload Scheduler installation directory. To use these latest definitions, you must merge the functions of your current FINAL and FINALPOSTREPORTS job streams with the syntax of your new FINAL and FINALPOSTREPORTS job streams.

Important: The definitions of the FINAL and FINALPOSTREPORTS job streams in `<`*`TWA_HOME`*`>/TWS/config/Sfinal` are defined on an extended agent that might not be defined in the new environment. If you are planning to use the old definitions to replace the new ones using the composer command `composer replace`, you must either change the workstation on which the jobs are defined to an existing one, or you must create a new extended agent where the jobs inside the *Sfinal* are defined.

Complete the following procedure:

1. Depending on your situation, edit your current final job streams and customize the new final job streams as follows:

   If you had customized job streams called FINAL and FINALPOSTREPORTS in your database:

   a. Extract the definitions from the current FINAL and FINALPOSTREPORTS job streams file by using composer.
   b. Use a text editor to edit your customized FINAL and FINALPOSTREPORTS job streams.
   c. Merge the job streams with file `<`*`TWA_HOME`*`>/TWS/config/Sfinal` so that the new FINAL and FINALPOSTREPORTS job streams have the same customization as your customized final job

streams plus the new required attributes provided by the new FINAL and FINALPOSTREPORTS job streams.

d. Save your new FINAL and FINALPOSTREPORTS job streams by using composer.

If you had customized final job streams called something other than FINAL and FINALPOSTREPORTS in your database:

a. Extract the definitions from your customized final job stream files by using composer.
b. Use a text editor to edit your customized final job stream files.
c. Merge the job streams with file `<TWA_HOME>/TWS/config/Sfinal` so that the new FINAL and FINALPOSTREPORTS job streams have the same customization as your customized final job streams plus the new required attributes provided by the new FINAL and FINALPOSTREPORTS job streams.
d. Save these new final job streams so that they have the same names as your current customized final job streams by running the command composer replace.

If you had final job streams called something other than FINAL and FINALPOSTREPORTS in your database, but they are not customized:

a. Make a copy of file `<TWA_HOME>/TWS/config/Sfinal`.
b. Edit this copy and rename the FINAL and FINALPOSTREPORTS parameters with the actual names.
c. Run the command composer replace.

If you had final job streams called FINAL and FINALPOSTREPORTS in your database, but they are not customized:

Run the command composer replace <TWA_HOME>/TWS/config/Sfinal.

If you had final job streams called FINAL and FINALPOSTREPORTS but they are in DRAFT in your database:

Run the command composer replace and, after the upgrade, change these job streams into the DRAFT status again.

2. After you customized the new final job streams, you must delete your current final job stream instances (**conman** cancel sched command) and submit the new final job stream instances (**conman** sbs sched command).

During the upgrade, JnextPlan is overwritten even if you customized it. The existing JnextPlan is backed up and renamed to:

On Windows™ operating systems:

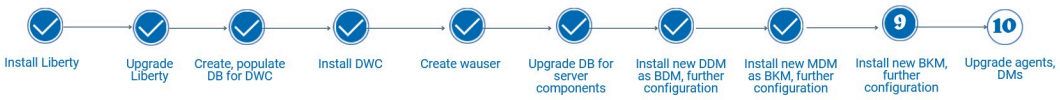
JnextPlan.cmd.bk

On UNIX™ and Linux™ operating systems:

JnextPlan.bk

Installing a new backup master domain manager

Upgrading your old master domain manager, which is now your current backup master domain manager to the latest product version level.

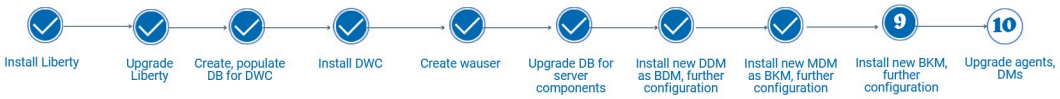


Now that you have a new master domain manager installed at the latest product version level, you can upgrade your old, previous version master domain manager, which is currently your backup master domain manager, to the latest product version to become the new backup master domain manager. You do this by installing a new backup master domain manager. Ensure you specify the same user as the one specified for the master domain manager.

Note: If you want to minimize the number of workstations required, you can install the new backup master domain manager on the same workstation where your old master domain manager was running. Ensure you stop any running processes related to the previous product version before installing the new backup master domain manager..

Installing a new backup master domain manager

Installing the new backup master domain manager



Before beginning the installation, ensure you have converted the certificates, as described in Converting default certificates *(on page 223)*.

You can perform a typical installation, as described in the following scenario, or you can customize the installation parameters, as described in FAQ - master domain manager and backup master domain manager customizations *(on page 74)*.

For more information about all **serverinst** parameters and default values, see Server components installation - serverinst script *(on page 310)*.

The procedure to install the backup master domain manager is exactly the same as installing a master domain manager. The backup master domain manager is installed on a workstation different from the master domain manager and points to its local WebSphere Application Server Liberty Base installation. IBM® Workload Scheduler detects the presence of an existing master domain manager in the environment and proceeds to install a backup master domain manager.

The IBM® Workload Scheduler administrator installs the master domain manager. The following information is required:

Table 21. Required information

Required information for performing the upgrade

Command parameter	Information type	Provided in..
IBM® Workload Scheduler information		
--wauser	IBM® Workload Scheduler administrative user name	Creating the IBM Workload Scheduler administrative user *(on page 239)*
--wapassword	IBM® Workload Scheduler administrative user password	
WebSphere Application Server Liberty Base information		
--wlpdir	WebSphere Application Server Liberty Base installation directory	Installing WebSphere Application Server Liberty *(on page 225)*
Security information		
--sslkeysfolder	location of converted certificates	Converting default certificates *(on page 223)*
--sslpassword	password of converted certificates	Converting default certificates *(on page 223)*

You can run the **serverinst** command specifying a typical set of parameters. In this case, default values are used for all remaining parameters. For more information about all **serverinst** parameters and default values, see Server components installation - serverinst script *(on page 310)*.

A properties file named `serverinst.properties` is available if you do not want to type parameters in the command line. This is expecially useful if you need to specify many parameters or if you want to reuse the file for several installations. The file is located in *`image_location`*`/TWS/`*`interp_name`*.

You can specify values in the properties file, type them in the command line, or use both methods. If a parameter is specified both in the properties file and in the command line, the command line value takes precedence.

If you need to modify any of the default values, edit the `serverinst.properties` file, but do not modify the `serverinst.template` file located in the same path.

To install the backup master domain manager, perform the following steps:

1. Log in to the workstation where you plan to install.
2. Browse to the folder where the `serverinst` command is located in *`image_location`*`/TWS/`*`interp_name`*.
3. Start the installation specifying a typical set of parameters. In this case, default values are used for all remaining parameters:

   On Windows operating systems

   ```
   cscript serverinst.vbs --acceptlicense yes --rdbmstype <db_type>
       --dbhostname <db_hostname> --dbport <db_port> --dbname <db_name>
       --dbuser <db_user> --dbpassword <db_password> --wauser <wa_user>
       --wapassword <wa_password> --wlpdir <Liberty_installation_dir>\wlp
       --sslkeysfolder <certificate_files_path> --sslpassword <keystore_truststore_password>
   ```

   On UNIX operating systems

   ```
   serverinst.sh --acceptlicense yes --rdbmstype <db_type>
           --dbhostname <db_hostname> --dbport <db_port> --dbname <db_name>
           --dbuser <db_user> --dbpassword <db_password> --wauser <wa_user>
           --wapassword <wa_password> --wlpdir <Liberty_installation_dir>/wlp
           --sslkeysfolder <certificate_files_path> --sslpassword
    <keystore_truststore_password>
   ```

   where

   acceptlicense

   Specify **yes** to accept the product license.

   rdbmstype|-r *rdbms_type*

   The database type. Supported databases are:
   - DB2
   - ORALE
   - MSSQL

   This parameter is optional. The default value is **db2**.

   dbhostname *db_hostname*

   The host name or IP address of database server.

   dbport *db_port*

   The port of the database server.

   dbname *db_name*

   The name of the IBM® Workload Scheduler database.

   dbuser *db_user*

   The user that has been granted access to the IBM® Workload Scheduler tables on the database server.

   dbpassword *db_password*

   The password for the user that has been granted access to the IBM® Workload Scheduler tables on the database server. Special characters are not supported.

   wauser *user_name*

   The user for which you are installing IBM Workload Scheduler.

wapassword ***wauser_password***

The password of the user for which you are installing IBM Workload Scheduler.

On Windows operating systems

Supported characters for the password are alphanumeric, dash (-), underscore (_) characters, and ()|?*~+.@!^

On UNIX operating systems

Supported characters for the password are any alphanumeric, dash (-), underscore (_) characters, and ()|?=*~+.

wlpdir

The path where WebSphere Application Server Liberty Base is installed.

--sslkeysfolder ***keystore_truststore_folder***

The name and path of the folder containing certificates in PEM format. The installation program automatically processes the keystore and truststore files using the password you specify with the **--sslpassword** parameter. The folder must contain the following files:

- **ca.crt**

  The Certificate Authority (CA) public certificate. Note that if certificates being installed are part of a chain consisting of 3 or more certificates (one Root CA, followed by one or more Intermediate CAs, followed by the end user certificate), then this file must contain the Root CA certificate only. Any Intermediate CA certificates must be stored in the `additionalCAs` subfolder, which therefore becomes a mandatory subfolder. Each Intermediate CA must be stored in the `additionalCAs` subfolder in its own file.

  Note: From V10.2.3, if certificates being installed are part of a chain, the ca.crt can contain also the intermediate CAs. In this case, it must begin with one or more intermediate CA certificates and end with the Root ca.

- **tls.key**

  The private key of the end user certificate for the instance to be installed.

- **tls.crt**

  The public part of the previous key, that is the end user certificate.

For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).

You can optionally create a subfolder to contain one or more `*.crt` files to be added to the server truststore as trusted CA, whose name must be `additionalCAs`. This can be used for example to add to the list of trusted CAs the certificate of the LDAP server or DB2 server. Additionally, you can store here any intermediate CA certificate to be added to the truststore. The subfolder must be named **additionalCAs**. Note that if the end user certificate being installed in the instance is part of a chain consisting of 3 or more certificates (one Root CA, followed by one or more Intermediate CAs, followed by the end user certificate), then the Intermediate CAs certificates must be stored in the `additionalCAs` subfolder, which therefore becomes a mandatory subfolder. Each Intermediate CA must be stored in the `additionalCAs` subfolder in its own file.

For further information about how to generate custom certificates, see Managing certificates using Certman *(on page)*.

--sslpassword ***ssl_password***

The password for the custom certificates and the path to the folder containing certificates in PEM format with the **sslkeysfolder** parameter.

For more information, see sslkeysfolder *(on page 316)*.

You can optionally encrypt the password using the secure script. For more information, see Optional password encryption - secure script *(on page 300)*.

4. To verify that the installation completed successfully, browse to the directory where you installed the backup master domain manager and type the following commands:

```
. ./tws_env.sh
```

```
optman ls
```

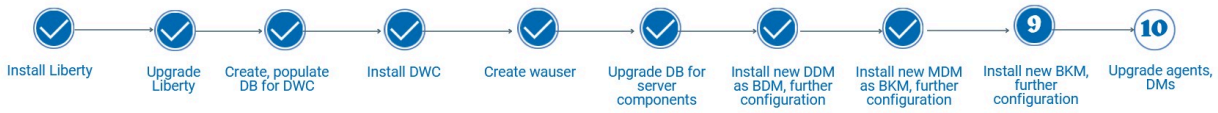
This command lists the IBM® Workload Scheduler configurations settings and confirms that IBM® Workload Scheduler installed correctly.

You have now successfully installed the backup master domain manager and it is inserted in the next production plan. To have the backup domain manager added immediately to the production plan, run

```
JnextPlan -for 0000
```

You can now proceed to Ensuring communication in your environment *(on page 259)*.

Ensuring communication in your environment



Security is enabled by default starting from version 10.1, but is usually not configured in most back-level environments. If security is not configured in your current environment, perform the following steps to ensure all IBM® Workload Scheduler components can communicate correctly:

Most 9.4 environments are not configured with SSL, which is enabled by default starting from version 10.1. To ensure communication between all components, perform the following steps:

1. On the backup master domain manager at version 10.2.3, stop WebSphere Application Server Liberty, as described in Application server - starting and stopping *(on page)*.
2. Browse to the following paths:

   on Windows operating systems

   `TWS\broker\config`

   on UNIX operating systems

   `TWS/broker/config`
3. Set the **Broker.Workstation.PortSSL** property to `false` in the `BrokerWorkstation.properties` file.
4. Start WebSphere Application Server Liberty on the backup master domain manager at version 10.2.3, as described in Application server - starting and stopping *(on page)*.
5. Run the following commands on the back-level master domain manager:
   a. `optman chg cf = ALL`

      This command changes the **enCarryForward** option so that all incomplete job streams are carried forward.

   b. `JnextPlan -for 0000 -noremove`

      This command extends the production plan without removing successfully completed job stream instances.

   c. `optman chg cf = <original value>`

      This command returns the **enCarryForward** option to its original value.

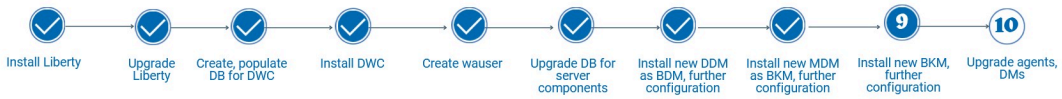
The new backup master domain manager can now communicate with the back-level network.

If you want to switch the new backup master domain manager to master, stop the broker on the back-level master domain manager, and switch it to master domain manager.

What to do next: You can now optionally proceed to Uninstalling the back-level backup master domain manager *(on page 260)*.

Uninstalling the back-level backup master domain manager

Procedure to uninstall the back-level backup master domain manager



1. Ensure that the user running the process has the following authorization requirements:

   Windows™ operating systems

   If you set the Windows User Account Control (UAC), your login account must be a member of the Windows™ **Administrators** group or domain administrators with the right, **Act as Part of the Operating System**.

   If you set the Windows User Account Control (UAC) on the workstation, you must run the installation as **administrator**.

   UNIX™ and Linux™ operating systems

   If the component was installed with root privileges, **root** access is required. If you performed a **no-root installation**, specify the same user used for installing the component.

2. Ensure that all IBM Workload Scheduler processes, services and the WebSphere Application Server Liberty process are stopped, and that there are no active or pending jobs. For information about stopping the processes and services see Starting and stopping processes on a workstation *(on page)*.

To uninstall a backup master domain manager, perform the following steps:

1. To uninstall the backup master domain manager, you must first remove it from the plan. Set the workstation running the backup master domain manager to `ignore`, using either the `composer mod cpu workstation_name>` command or from the Dynamic Workload Console.
2. Run JnextPlan to generate the new production plan so that the backup master domain manager is removed from the plan.
3. Run the uninstall script.

   a. Change directory using the following command:

   ```
   cd TWA_home>/TWS/tws_tools
   ```

   b. Run the uninstallation process by running the script as follows:

   Windows™ operating systems

   ```
   cscript uninstall.vbs --prompt no --wauser user_name>
   ```

   UNIX™ and Linux™ operating systems

   ```
   ./uninstall.sh --prompt no --wauser user_name
   ```

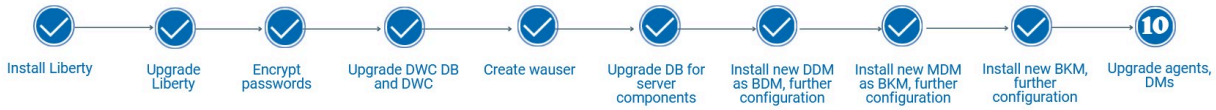
   where, *user_name>* represents the user for which you want to uninstall the backup master domain manager. The procedure runs without prompting the user to confirm the uninstallation.

4. Run JnextPlan to update the plan with the changes.

You can now proceed to Upgrading agents and domain managers *(on page 261)*.

Upgrading agents and domain managers

There are several methods you can choose from to upgrade your domain managers and



agents.

The agent upgrade can be performed with minimal impact to scheduling activities. The agents are stopped for the shortest time necessary to perform the maintenance. Any active agent command-line interfaces and processes, such as conman, composer, netman, mailman, and batchman, to name a few, continue running. Any jobs already running when the upgrade process begins, continue to run as planned, however, no new jobs begin execution during this time. Once the upgrade is complete, the agent is restarted and quickly reconnects with its jobs. Any jobs that were actively running before the upgrade that have not yet completed, continue to run, and any jobs that successfully finished running during the upgrade procedure report a successful job status. An automatic backup and restore feature is in place in case of failure.

Because domain managers are agents, they are upgraded using the procedures described in this section.

If you choose to upgrade your environment top-down, then the agents get upgraded progressively after you have upgraded the master domain manager and its backup. This means that new features and enhancements are not available on all of your agents at the same time. If, instead, you choose to upgrade your environment bottom-up, then the agents are upgraded first, and new features and enhancements become available after the master domain manager and its backup have been upgraded.

Important: After upgrading your fault-tolerant agents, it might be necessary to manually update the security file on the fault-tolerant agents in your environment to add access to folders for all of the scheduling objects that can be defined or moved into folders. These updates are especially important if you plan to use the command line on the fault-tolerant agents to perform operations on the objects in folders. See Updating the security file *(on page)* for more information.

You can choose to upgrade your agents using any of the following methods:

twsinst script

A single line command that checks if processes or a command line is running before it starts. It saves disk space and RAM because it is not Java-based. See Upgrade procedure *(on page 201)* and Upgrading agents on IBM i systems *(on page 205)*

Centralized agent update

Upgrade or update multiple fault-tolerant agent and dynamic agent instances at the same time. Download the fix pack installation package, or the eImage upgrade package to the master domain manager and then either run the installation on multiple agent instances or schedule the installation by creating and submitting a job to run. This upgrade method is not supported on z-centric agent instances. See Centralized agent update *(on page 208)*.

HCL BigFix

Upgrade IBM® Workload Scheduler agents using HCL BigFix analyses and fixlets. You can choose to schedule the upgrade or you can run it immediately. See Upgrading agents using HCL BigFix *(on page 216)*.

For a list of supported operating systems and requirements, see the System Requirements Document at IBM Workload Scheduler Detailed System Requirements.

When the upgrade procedure has completed successfully, the backup instance is deleted.

Note: The `localopts` file is not modified during the agent upgrade process. The file generated by the upgrade process is saved to the `/config` directory to maintain your custom values, if any. You can then merge the two files with your customized values and save the resulting file in the following path:

On Windows operating systems

<*TWA_home*>\TWS

On UNIX operating systems

<*TWA_DATA_DIR*>

When upgrading dynamic agents featuring both a local and a remote gateway, ensure you either upgrade the agent first and then the gateway or upgrade both at the same time.

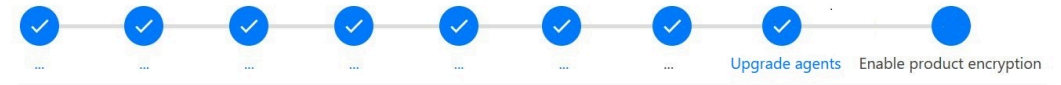
Related information

Upgrading the database for the server components *(on page 239)*
Enabling product encryption after upgrading *(on page 262)*

Enabling product encryption after upgrading

Enabling product encryption after upgrading from a version earlier than 10.1.

If you are upgrading from a version earlier than version 10.1, you can optionally enable encryption for key product files by performing the following steps on the master domain manager and on each agent in the environment:



1. Generate a new key by running the following keytool command:

```
./keytool -genseckey -alias new_alias_name -keyalg AES -keysize 256
-storepass encrypt_keystore_pwd_in_clear -storetype PKCS12 -keystore encrypt_keystore_file
```

2. Create the stash file containing a password encoded in base64. You can store the file in a path of your choice.
3. Add the following keys in the `localopts` file:

   encrypt keystore file *file_name*

   The path to the keystore PKCS12 file, containing the AES-256 or AES-128 key.

   encrypt keystore pwd *password*

   The path to the keystore stash file.

   encrypt label

   The label you assign to the new key in the keystore. This property is case insensitive.

   Consider the following example of the modifications to the `localopts` file:

```
encrypt keystore file ="/opt/wa/TWA/TWS/ssl/key.p12"
encrypt keystore pwd ="/opt/wa/TWA/TWS/ssl/key.sth"
encrypt label ="myalias"
```

   where

   encrypt keystore file

   corresponds to the **-keystore** *encrypt_keystore_file* parameter in the command provided in step 1.

   encrypt keystore pwd

   corresponds to the path of the stash file created in step 2.

   encrypt label

   corresponds to the **-alias** *new_alias_name* parameter in the command provided in step 1.

The current Symphony plan keeps using the previous key. To apply the new setting to the Symphony plan, run a JnextPlan command. The message boxes are encrypted immediately and the `useropts` file is encrypted as soon as you save the `localopts` file and launch a CLI command. Key product files are now encrypted with the new key.

Enabling API Key authentication after upgrading

Enabling API Key authentication after upgrading from v 10.*x*.*x* or v 9.5.*x* to 10.2.*x*.

In previous versions of the product, both in fresh and upgrade installation, it was not necessary to add the server public certificate to its truststore. With the new API Key feature, which is implemented in version 10.1 Fix Pack 1 and later, the generated JWT is signed with the server private key. When the JWT is received by the server to authenticate a user, the public key associated with the private key used for signing is not present in the truststore and cannot be used. As a result, the authentication of that user is blocked.

To solve the problem, in fresh installations the server public key is automatically added to its trustore.

When you are upgrading from v 10.*x*.*x* or v 9.5.*x* to 10.2.*x*, run the following commands on the master domain manager:

1. ```
   keytool -exportcert -keystore
    $WA_DATADIR/usr/servers/engineServer/resources/security/TWSServerKeyFile.p12
   -storepass password -storetype pkcs12 -file /tmp/tls.crt -alias server -noprompt
   ```
2. ```
   keytool -importcert -keystore
    $WA_DATADIR/usr/servers/engineServer/resources/security/TWSServerTrustFile.p12
   -storepass password -storetype pkcs12 -file /tmp/tls.crt -alias mpjwtkey -noprompt
   ```
3. Edit the value of the **mp.jwt.trust.key** variable from the **twstrustkey** to **mpjwtkey** in the `jwt_variables.xml` file located inside the WebSphere Application Server Liberty Base `overrides` folder. For more information about templates, see Configuring IBM Workload Scheduler using templates *(on page)*.

If you do not remember what the public certificate alias is called, run the following command to retrieve the list of certificates within the keystore:

```
keytool -list -keystore $WA_DATADIR/usr/servers/engineServer/resources/security/TWSServerKeyFile.p12
-storepass password -storetype pkcs12
```

Upgrading when there are corrupt registry files

If you have tried to upgrade a stand-alone, fault-tolerant agent (an agent that is not shared with other components or does not have the connector feature) and received an error message that states that an instance of IBM Workload Scheduler cannot be found, this can be caused by a corrupt registry file. It is possible to upgrade a stand-alone, fault-tolerant agent that has corrupt registry files without having to reinstall the product. IBM Workload Scheduler has a recovery option you can run to re-create the necessary files. You can also use this option when upgrading nodes in clusters, where the node on which you want to perform the upgrade is not available or is in an inconsistent state. The recovery option re-creates the registry files and the Software Distribution information without having to reinstall the complete product.

You can run the recovery option using the **twsinst** script.

Re-creating registry files using twsinst

To re-create the registry files while upgrading an agent by using the **twsinst** script, from the directory that contains the IBM Workload Scheduler agent eImage, run **twsinst** using the synopsis described below.

Synopsis:

On Windows™ operating systems:

Show command usage and version

```
twsinst -u | -v
```

Upgrade an instance

```
twsinst -update -uname user_name -password password
..-acceptlicense yes|no
  [-domain user_domain]
```

```
  [-recovInstReg true]
  [-inst_dir install_dir]
```

Example

```
cscript twsinst -update -uname twsuser -password twspassword
-acceptlicense yes -inst_dir "C:\Program Files\IBM\TWA"
-recovInstReg true
```

On UNIX™ and Linux™ operating systems

Show command usage and version

```
./twsinst -u | -v
```

Upgrade an instance

```
./twsinst -update -uname user_name
..-acceptlicense yes|no
..[-inst_dir install_dir
..[-recovInstReg true]]
```

Example

```
./twsinst -update -uname twsuser -inst_dir /opt/IBM/TWA
-acceptlicense yes -recovInstReg true
```

For information about the **twsinst** parameters, see Upgrade procedure *(on page 201)*.

Upgrading in a mixed-version environment when using default certificates

Upgrading in a mixed-version environment when using default certificates

If your environment contains components, such as agents, Dynamic Workload Console, dynamic domain managers, and so on, at various version levels and you use default certificates, ensure certificates across the environment are consistent.

For example, you might need to install an agent at version 10.2.*x*, and connect it to a back-level master domain manager.

If you are using default certificates, you need to convert them to the new format and make them available to all components before you start the upgrade, as described in the following steps:

1. Set the IBM® Workload Scheduler environment, as described in Setting the environment variables *(on page 138)*.
2. To ensure the keytool and openssl commands start correctly on all operating systems, browse to the folder where the keytool and openssl commands are located and launch the commands as follows:

   ```
   cd <TWS_DIR>/JavaExt/jre/jre/bin
   ```

   ```
   ./keytool -importkeystore -srckeystore TWSServerKeyFile.jks -destkeystore
   <path_of_extracted_certs>/server.p12 -deststoretype pkcs12
   ```

   ```
   cd <TWS_DIR>/tmpOpenSSL64/1.1/bin/openssl
   ```

   ```
   ./openssl pkcs12 -in <path_of_extracted_certs>/server.p12 -out
   <path_of_extracted_certs>/tls.tot
   ```

   The location of the `TWSServerKeyFile.jks` varies depending on the IBM® Workload Scheduler version you have currently installed, as follows:

   versions 9.5 and later

   TWA_DATA_DIR`/usr/servers/engineServer/resources/security`

   versions 9.4 and earlier

   TWA_home`/WAS/TWSProfile/etc`

3. Open the `tls.tot` file with any text editor.
4. From the `tls.tot` file, copy the private key to a new file named `tls.key`.
   The `tls.key` file must be structured as follows:

```
----BEGIN ENCRYPTED PRIVATE KEY----
<private_key>
----END ENCRYPTED PRIVATE KEY----
```

Note: Insert a carriage return after each key, so that an empty line is inserted after each key.

5. From the `tls.tot` file, copy the public key to a new file named `tls.crt`.
   The `tls.crt` file must be structured as follows:

```
----BEGIN CERTIFICATE----
<public_key>
----END CERTIFICATE----
```

Note: Insert a carriage return after each key, so that an empty line is inserted after each key.

6. Copy the contents of the `tls.crt` file into a new file named `ca.crt`. If you want to upgrade a dynamic domain manager, also copy the contents of the `tls.crt` file into another new file named `jwt.crt`.
7. Create a file named `tls.sth` containing the passphrase you have specified for creating the .`p12` certificate in step 2 *(on page 264)*, encoded in `base64` format. To create the `tls.sth` file, use the following command:

```
./secure -password your_password -base64 e -out
<path_of_extracted_certs>/tls.sth
```

   If you are using a version earlier than 10.*x*, you can find the secure script in the installation package of the 10.2.3 version you are upgrading to. You can launch the script from on of the following paths:

   master domain manager and agent

```
<10.2.3_extracted_image_dir>/TWS/<interp>/Tivoli_LWA_<interp>/TWS/bin
```

   Dynamic Workload Console

```
<10.2.3_extracted_image_dir>/DWC/<interp>/bin
```

   where

   <interp>

   is the operating system you are installing on

   As an alternative, you can use the following command on UNIX workstations:

```
echo -n "passwordToEncode" | base64 >> tls.sth
```

8. Browse to the GSKit folder and extract the client certificates from the *TWA_DATA_DIR*`/ssl/GSKit` folder by running the following commands, depending on the IBM® Workload Scheduler version you have currently installed:

```
cd <TWS_DIR>/tmpGSKit64/8/bin
```

   versions 9.5 and later

```
./gsk8capicmd_64 -cert -extract -db <TWA_DATA_DIR>/ssl/GSKit/TWSClientKeyStore.kdb
-stashed -label client -target <path_of_extracted_certs>/client.crt
```

   versions 9.4 and earlier

```
./gsk8capicmd_64 -cert -extract -db <TWS_DIR>/ssl/GSKit/TWSClientKeyStore.kdb
-stashed -label client -target <path_of_extracted_certs>/client.crt
```

9. Create a folder named `additionalCAs` in the folder where you extracted the certificates and move the `client.crt` file created in step 8 *(on page 265)* to the `additionalCAs` folder.
10. Insert the `client.crt` in the `additionalCAs` folder when providing the certificates to the installation script with the **sslkeysfolder** parameter.

11. Assign the correct permissions (755) and ownerships to extracted certificates, as follows:

```
chmod -R 755 <path_of_extracted_certs>
```

You have now converted the certificates to the required PEM format.

You can now use the new default certificates for installing or upgrading IBM® Workload Scheduler components, as follows:

If your master domain manager is at least at 10.1 FP1 level

you can copy the certificates you converted with the above procedure to the `/depot` folder on the master domain manager and install or upgrade dynamic agents and fault-tolerant agents specifying the **wauser** and **wapassword** parameters. For all remaining components, copy the certificates locally and launch the installation or upgrade specifying the **sslkeysfolder** and **sslpassword** parameters.

If your master domain manager is at a version earlier than 10.1 FP1 level

copy the certificates you converted with the above procedure locally on all components and launch the installation or upgrade specifying the **sslkeysfolder** and **sslpassword** parameters.

For more information about all installation and upgrade parameters, see the serverinst, dwcinst, and twsinst scripts in Reference *(on page 300)*.

Chapter 3. Updating containers

Updating the container configuration parameters.

To change the container configuration parameters or to obtain the latest version of a container, an update is required.

Complete the following procedure to update a Docker container:

1. Log in to My IBM Container software library with your IBMid and password.
2. From the Container software library, click Copy key to copy the Entitlement key.
3. Run the following command to log into the IBM Entitled Registry:

   ```
   docker login -u cp -p <your_entitlement_key> cp.icr.io
   ```

4. Manually update the compose file by modifying the *image* name if docker-compose does not reference the version to which you want to update.
5. Launch the "`docker-compose up -d`" command.

Note:

- Launching the "`docker-compose up -d`" command, the container is restarted and the database schema is automatically updated. If you are planning to update both the IBM Workload Automation server MDM and BKM, ensure that you run the command for one component at a time. To avoid database conflicts, start the second component only when the first component has completed successfully.
- In a Docker environment, if your server component uses a timezone different from the default timezone, then to avoid problems with the FINAL job stream, you must update `MAKEPLAN` within the DOCOMMAND, specifying the **timezone** parameter and value. For example, if you are using the America/Los Angeles timezone, then it must be specified as follows:

  ```
  $JOBS

  WA_WA-SERVER_XA#MAKEPLAN
  DOCOMMAND "TODAY_DATE=`${UNISONHOME}/bin/datecalc today pic YYYYMMDD`; ${UNISONHOME}/MakePlan
   -to `${UNISONHOME}/bin/datecalc ${TODAY_DATE}070
  0 + 1 day + 2 hours pic MM/DD/YYYY^HHTT` timezone America/Los_Angeles"
  STREAMLOGON wauser
  DESCRIPTION "Added by composer."
  TASKTYPE OTHER
  SUCCOUTPUTCOND CONDSUCC "(RC=0) OR (RC=4)"
  RECOVERY STOP
  ```

Only the following parameters can be modified with the update:

- DB_TYPE
- DB_HOSTNAME
- DB_PORT
- DB_NAME
- DB_TS_NAME
- DB_TS_PATH
- DB_LOG_TS_NAME
- DB_LOG_TS_PATH
- DB_PLAN_TS_NAME
- DB_PLAN_TS_PATH
- DB_TEMP_TS_NAME
- DB_SBSPACE
- DB_USER
- DB_ADMIN_USER
- DB_SSL_CONNECTION
- WA_PASSWORD

- DB_ADMIN_PASSWORD
- DB_PASSWORD
- SSL_KEY_FOLDER
- SSL_PASSWORD

Updating containers when using default certificates

Updating the container configuration parameters when using default certificates.

Before you begin:

Modify the certificates as explained in the following procedure:

1. Access the server container.
2. Open the `localopts` file and check the certificates path in the following section:

```
SSL key   ="/home/wauser/wadata/FTAcert/TWSClient.key"
SSL certificate  ="/home/wauser/wadata/FTAcert/TWSClient.cer"
SSL key pwd  ="/home/wauser/wadata/FTAcert/password.sth"
SSL CA certificate  ="/home/wauser/wadata/FTAcert/TWSTrustCertificates.cer"
SSL random seed  ="/home/wauser/wadata/FTAcert/TWS.rnd"
```

3. Exit the server container.
4. Copy all the certificates in a local directory by launching the following command: `docker cp`.
5. Rename the certificates as follows:

```
tls.key
tls.crt
tls.sth
ca.crt
tls.rnd
```

6. Ensure that in the docker compose.yaml file you have the following parameters for server, console, and agent components:

```
SSL_PASSWORD= default
SSL_KEY_FOLDER= <cert_directory>
```

   where

   <cert_directory> is the path of the directory where you saved the certificates.

7. Modify the volume `<path_on_host_containing_certs>:/opt/wautils/certs` with the path of the directory that contains your certificates at the place of `<path_on_host_containing_certs>`.

About this task:

To change the container configuration parameters or to obtain the latest version of a container, an update is required.

Complete the following procedure to update a Docker container:

1. Log in to My IBM Container software library with your IBMid and password.
2. From the Container software library, click Copy key to copy the Entitlement key.
3. Run the following command to log into the IBM Entitled Registry:

```
docker login -u cp -p <your_entitlement_key> cp.icr.io
```

4. Manually update the compose file by modifying the *image* name if docker-compose does not reference the version to which you want to update.
5. Launch the "`docker-compose up -d`" command.

Note:

- Launching the "`docker-compose up -d`" command, the container is restarted and the database schema is automatically updated. If you are planning to update both the IBM Workload Automation server MDM and BKM, ensure that you run the command for one component at a time. To avoid database conflicts, start the second component only when the first component has completed successfully.
- In a Docker environment, if your server component uses a timezone different from the default timezone, then to avoid problems with the FINAL job stream, you must update `MAKEPLAN` within the DOCOMMAND, specifying the **timezone** parameter and value. For example, if you are using the America/Los Angeles timezone, then it must be specified as follows:

```
$JOBS

WA_WA-SERVER_XA#MAKEPLAN
DOCOMMAND "TODAY_DATE=`${UNISONHOME}/bin/datecalc today pic YYYYMMDD`; ${UNISONHOME}/MakePlan
 -to `${UNISONHOME}/bin/datecalc ${TODAY_DATE}070
0 + 1 day + 2 hours pic MM/DD/YYYY^HHTT` timezone America/Los_Angeles"
STREAMLOGON wauser
DESCRIPTION "Added by composer."
TASKTYPE OTHER
SUCCOUTPUTCOND CONDSUCC "(RC=0) OR (RC=4)"
RECOVERY STOP
```

Only the following parameters can be modified with the update:

- DB_TYPE
- DB_HOSTNAME
- DB_PORT
- DB_NAME
- DB_TS_NAME
- DB_TS_PATH
- DB_LOG_TS_NAME
- DB_LOG_TS_PATH
- DB_PLAN_TS_NAME
- DB_PLAN_TS_PATH
- DB_TEMP_TS_NAME
- DB_SBSPACE
- DB_USER
- DB_ADMIN_USER
- DB_SSL_CONNECTION
- WA_PASSWORD
- DB_ADMIN_PASSWORD
- DB_PASSWORD
- SSL_KEY_FOLDER
- SSL_PASSWORD

Chapter 4. FAQ - Upgrade procedures

A list of questions and answers related to upgrade procedures:

Q:How do I upgrade a component that was originally installed without SSL configuration?

A: To configure SSL attributes, perform the following steps:

1. Set the **security_level** parameter to **force_enabled** in the workstation definition and the **secureaddr** parameter to the secure port, as described in Configuring SSL attributes *(on page)*.
2. Set the **nm SSL full port** parameter to the value of the secure port in the `localopts` file. For more information, see Localopts details *(on page)*

Q: How do I upgrade a component that was installed with default certificates?

A: Define the **JKS_SSL_PASSWORD** environment variable as described in Enhanced security for default certificates *(on page)*. For the full upgrade procedure, see Upgrading *(on page 154)*. If you are using default certificates and want to install a new component to be connected to a back-level master, see Upgrading in a mixed-version environment when using default certificates *(on page 264)*.

Q: What happens if I do not remember the password for the default certificates?

A: Before starting the upgrade, test the passwords for the certificates using the following keytool commands:

- ```
  keytool -list -keystore TWSServerTrustFile.jks
  -storepass my_password
  ```
- ```
  keytool -list -keystore TWSServerKeyFile.jks
  -storepass my_password
  ```

Q: The upgrade failed because the password I provided for the certificates in the JKS_SSL_PASSWORD variable is incorrect. How can I recover from this error?

A. Before restarting the upgrade, perform the following steps:

1. Retrieve and test the password for the certificates, as described in Q: What happens if I do not remember the password for the default certificates? *(on page 270)*
2. Restore the previous version of the `ita.ini` file.
3. Restart the upgrade.

Q: My environment is FIPS compliant. What happens if I upgrade to version 10.2.3?

A: Version 10.2.3 does not support FIPS. If you want to upgrade to this version, your environment will no longer be FIPS compliant. A new optional parameter named **enablefips** is available in the serverinst and twsinst scripts to check FIPS settings before you upgrade. This is because you need to be aware that by upgrading, your environment will no longer be FIPS compliant.

Upgrade scenarios vary depending on your upgrade path, as follows:

If you are upgrading from version 10.2.1, or later

FIPS is already disabled by default in this version. If do not specify the **enablefips** parameter or you set it to `false`, the upgrade proceeds. If you set the **enablefips** parameter to `true`, the upgrade stops with an error message and you have to set **enablefips** to `false` to proceed.

If you are upgrading from a version earlier than 10.2.1

You can proceed in one of the following ways:

- Disable FIPS before upgrading by editing the following options in the configuration files:

  localopts

  set **SSL Fips enabled** to `no`

  ita.ini

set **fips_enable** to `no`

You can then proceed with the upgrade without specifying the **enablefips** parameter, which is set to `false` by default.

- Set the **enablefips** parameter to `false`. A warning message is displayed to inform you that FIPS is being disabled and the `localopts` and `ita.ini` files are automatically updated with the new FIPS configuration (the previous **SSL Fips enabled** option is removed and the new **SSL FIPS compliance** option is added and set to `no/false`) . The upgrade proceeds.

Can I install a backup master domain manager at version 10.2.3 in a back-level environment?

If you have a back-level environment, for example version 9.4, you can install a backup master domain manager at version 10.2.3, but it is recommended you check your security configuration.

Most 9.4 environments are not configured with SSL, which is enabled by default starting from version 10.1. To ensure communication between all components, see Ensuring communication in your environment *(on page 259)*

How can I get the dynamic agent installed on the new backup master domain manager to communicate with the back-level master domain manager?

In back-level environments, for example 9.4, SSL is not enabled by default and TLS version 1.2 needs to be enabled on the back-level master domain manager to enable communication. Perform the following steps on the back-level master domain manager, as described in Configuring TLS to the appropriate version *(on page 222)*.

For more information, see Switching from SSLv3 to TLSv1.2 and steps 2 and 3 in How to Run Composer on a 9.5 FTA Connecting to a 9.4 MDM

Part V. Moving your workload from an on-premises to a cloud environment

A quick procedure to move your workload from an on-premises to a cloud environment

Moving your workload from an on-premises to a cloud environment is a quick procedure which involves configuring SSL communication between your existing on-premises master domain manager and a new backup master domain manager on the cloud. You then switch permanently domain management capabilities from the on-premises master domain manager to the backup master domain manager on the cloud to shift your whole workload to the cloud. This procedure requires the on-premises master domain manager to be at Version 9.5 Fix Pack 3 or later.

At the end of the procedure, you will have switched your master domain manager to the cloud and set up your dynamic agents to work in SSL mode with the on-cloud master domain manager

This procedure applies to the following clusters:

Amazon Elastic Kubernetes Service (EKS)

For this cluster, you can use an ingress-type network or a load-balancer network. To specify which network type you want to use, set the relevant parameters in the `values.yaml` file. For detailed information, see the **Network enablement** section in IBM Workload Automation.

OpenShift

For this cluster, you can only use routes as network service. An OpenShift Container Platform route allows you to associate a service with an externally-reachable host name. This edge host name is then used to route traffic to the service. For more information, see the readmes available in Deploying IBM Workload Automation components on Red Hat OpenShift *(on page 123)*.

Note: On-premises fault-tolerant agents cannot connect to an on-cloud master domain manager.

On-premises side operations

Ensure the following conditions are met for your on-premises master domain manager:

- Version 9.5, Fix Pack 3 or later is installed.
- The port number used by the `netman` process to listen for communication from the dynamic domain manager (**brnetmanport**) is set to the default **41114** value.
- Ensure the SECURITYLEVEL attribute is set to `force`, or `force_enabled`. For more information about workstation definition parameters, see Workstation definition *(on page)*.

Perform the following operations on the on-premises side:

1. Set the IBM® Workload Scheduler environment variables:

   In UNIX®:

   - `. ./TWA_home/TWS/tws_env.sh` for Bourne and Korn shells
   - `. ./TWA_home/TWS/tws_env.csh` for C shells

   In Windows®:

   - `TWA_home\TWS\tws_env.cmd`

2. Configure your master domain manager for SSL communication using the modify command:

   ```
   composer modify ws your_master_domain_manager
   ```

a. In the **secureaddr** argument, define the port used to listen for incoming SSL connections, for example `31113` or another available port.
b. In the **securitylevel** argument, specify `enabled` to set the master domain manager to uses SSL authentication only if its domain manager workstation or another fault-tolerant agent below it in the domain hierarchy requires it.

See the following example:

```
CPUNAME your_mdm_name
  DESCRIPTION "MANAGER CPU"
  OS UNIX
  NODE your_IP_address TCPADDR 31111
  SECUREADDR 31113
  DOMAIN MASTERDM
  FOR MAESTRO
    TYPE MANAGER
    AUTOLINK ON
    BEHINDFIREWALL OFF
    SECURITYLEVEL ENABLED
    FULLSTATUS ON
END
```

For more information about the modify command, see modify *(on page)*. For more information about workstation properties, see Workstation definition *(on page)*.

3. Modify the `localopts` file to enable SSL communication, as follows:

   a. Browse to the *TWA_DATA_DIR* folder.

   b. Edit the following properties in the `localopts` file. See the following example:

```
nm SSL full port   =0
nm SSL port        =31113
SSL key            ="/install_dir/ssl/OpenSSL/TWSClient.key"
SSL certificate    ="/install_dir/ssl/OpenSSL/TWSClient.cer"
SSL key pwd        ="/install_dir/ssl/OpenSSL/password.sth"
SSL CA certificate  ="/install_dir/ssl/OpenSSL/TWSTrustCertificates.cer"
SSL random seed    ="/install_dir/ssl/OpenSSL/TWS.rnd"
```

where:

nm SSL port

Is the port used to listen for incoming SSL connections, when full SSL is not configured, for example `31113`.

For more information about the `localopts` file, see Setting local options *(on page)*.

4. If you have a dynamic domain manager in your environment, repeat steps 2 *(on page 272)* and 3 *(on page 273)* on the dynamic domain manager to have the dynamic domain manager function correctly with the on-cloud master domain manager. The dynamic domain manager stays in the on-premises environment.
5. If you want to use custom SSL certificates, edit the paths in the `localopts` file specifying the paths to the custom certificates and using the same names as the default certificates. For more information about secure connections, see Connection security overview *(on page)*, and specifically Extending communication scenarios to other server components *(on page)*.
6. Stop IBM® Workload Scheduler Batchman process by running this command:

```
conman stop
```

7. Stop IBM® Workload Scheduler Netman process by running this command:

```
conman shut
```

8. Restart IBM® Workload Scheduler processes by running these commands:

```
StartUp
```

```
conman start
```

9. You can optionally configure your on-premises fault-tolerant agents for communicating with the on-cloud master domain manager, by performing this procedure on each fault-tolerant agent.

Cloud-side operations

If you are using OpenShift, the connection between the on-premises master domain manager and the on-cloud backup master domain manager takes place through routes; therefore, it is recommended to use short names for namespaces, especially if the cluster name is long. This is because workstation host names cannot exceed 51 characters, therefore, the route must comply with this maximum character length.

Perform the following operations on the cloud side:

1. Download the latest product version. See
   - If you are using Amazon EKS, see IBM Workload Automation for information about downloading images, installing, and configuring the product.
   - If you are using OpenShift, see Deploying IBM Workload Automation components on Red Hat OpenShift *(on page 123)*.
2. Open the `values.yaml` file to configure a new server instance.

   If you want to deploy only a new server without the Agent and Console applications, set the **enableAgent** and **enableConsole** parameters to `false`.
3. Set the following database parameters to have the new server instance point the database of the on-premises master domain manager. These values must match the values defined for the on-premises master domain manager.

   ```
   db:
     adminUser: <admin_dbuser>
     hostname: <db_host>
     name: <db_name>
     port: <db_port>
     sslConnection: false
     tsName: null
     tsPath: null
     tsTempName: null
     tssbspace: null
     type: <db_type>
     usepartitioning: true
     user: <db_user>
   ```

   This automatically configures the on-cloud server as a backup master domain manager for the on-premises master domain manager.
4. Set the **server.enableSingleInstanceNetwork** parameter to `true` to create an additional load balancer for each server pod. This is used to connect the backup master domain manager inside the cluster with master domain manager outside the cluster. For more information about parameters, see the **Configuration Parameters** section in IBM Workload Automation.
5. To deploy the new server instance in a cloud environment, type:

   ```
   helm install -f values.yaml workload_automation_release_name workload/ibm-workload-automation-prod
    -n workload_automation_namespace
   ```

   where:

   workload_automation_release_name

   is the name of the release, for example `hwa`.

   When you deploy the backup master domain manager on the cloud, it is automatically configured as follows, in full SSL mode with the on-premises master domain manager:

   ```
   CPUNAME HWA-SERVER-0
    DESCRIPTION "FTA CPU"
    OS UNIX
    NODE hwa-waserver-0.hwa-test TCPADDR 31111
    SECUREADDR 443
    DOMAIN MASTERDM
    FOR MAESTRO
     TYPE FTA
     AUTOLINK ON
     BEHINDFIREWALL OFF
   ```

```
  SECURITYLEVEL FORCE_ENABLED
  FULLSTATUS ON
END
```

where

hwa-waserver-0.hwa-test

Is the name of the ingress-type network being configured, if you are using an ingress-type network for EKS.

If you are using a load-balancer network, the NODE parameter is automatically set to the IP address of the load balancer. For more information, see the **Network enablement** section in IBM Workload Automation.

If you are deploying on OpenShift, this parameter is automatically set to the OpenShift network route. For more information, see the readmes available in Deploying product components on Red Hat OpenShift, V4.*x*.

SECURITYLEVEL

Specifies the type of SSL authentication for the workstation. This parameter is automatically set to `force_enabled`, which means that the workstation uses SSL authentication for all of its connections to all target workstations which are set to this value. The workstation tries to establish a connection in FULLSSL mode and, if the attempt fails, it tries to establish an unsecure connection. For more information about workstation definition parameters, see Workstation definition *(on page)*.

In the same way, the `localopts` file of the backup master domain manager on the cloud is also automatically configured for SSL communication. See the following example:

```
nm SSL full port    =31113
#
nm SSL port       =0
#
SSL key  ="/home/wauser/wadata/FTAcert/TWSClient.key"
SSL certificate ="/home/wauser/wadata/FTAcert/TWSClient.cer"
SSL key pwd ="/home/wauser/wadata/FTAcert/password.sth"
SSL CA certificate ="/home/wauser/wadata/FTAcert/TWSTrustCertificates.cer"
SSL random seed ="/home/wauser/wadata/FTAcert/TWS.rnd"
```

6. To assign full control for all objects to the **wauser**, type the following command:

```
composer mod acl @
```

The following example shows the modified access control list:

```
ACCESSCONTROLLIST FOR ALLOBJECTS
  root FULLCONTROL
  twsuser FULLCONTROL
  wauser FULLCONTROL
END
```

```
ACCESSCONTROLLIST FOLDER /
  root FULLCONTROL
  twsuser FULLCONTROL
  wauser FULLCONTROL
END
```

Switching domain manager capabilities

Final steps to switch domain manager capabilities permanently

1. To switch the event processor, run the following command either on the master domain manager or backup master domain manager:

```
switcheventprocessor [folder/]workstation
```

For more information about the command, see switcheventprocessor *(on page)*.

2. To switch domain management capabilities, run the following command either on the master domain manager or backup master domain manager:

   ```
   switchmgr domain;newmgr
   ```

   For more information about the command, see switchmgr *(on page)*.
3. To make the switch permanent, edit from composer the definition of the previous master domain manager. See the following example and notice how the **TYPE** attribute changes from `MANAGER` to `FTA`.
   PREVIOUS DEFINITION

   ```
   CPUNAME your_mdm_name
     DESCRIPTION "MANAGER CPU"
     OS UNIX
     NODE your_IP_address TCPADDR 31111
     SECUREADDR 31113
     DOMAIN MASTERDM
     FOR MAESTRO
       TYPE MANAGER
       AUTOLINK ON
       BEHINDFIREWALL OFF
       SECURITYLEVEL ENABLED
       FULLSTATUS ON
   END
   ```

   NEW DEFINITION

   ```
   CPUNAME your_mdm_name
     DESCRIPTION "MANAGER CPU"
     OS UNIX
     NODE your_IP_address TCPADDR 31111
     SECUREADDR 31113
     DOMAIN MASTERDM
     FOR MAESTRO
       TYPE FTA
       AUTOLINK ON
       BEHINDFIREWALL OFF
       SECURITYLEVEL ENABLED
       FULLSTATUS ON
   END
   ```
4. To make the switch permanent, edit from composer the definition of the previous backup master domain manager. See the following example and notice how the **TYPE** attribute changes from `FTA` to `MANAGER`.
   PREVIOUS DEFINITION

   ```
   CPUNAME HWA-SERVER-0
     DESCRIPTION "FTA CPU"
     OS UNIX
     NODE hwa-waserver-0.hwa-test TCPADDR 31111
     SECUREADDR 443
     DOMAIN MASTERDM
     FOR MAESTRO
       TYPE FTA
       AUTOLINK ON
       BEHINDFIREWALL OFF
       SECURITYLEVEL FORCE_ENABLED
       FULLSTATUS ON
   END
   ```

   NEW DEFINITION

   ```
   CPUNAME HWA-SERVER-0
      DESCRIPTION "FTA CPU"
      OS UNIX
      NODE hwa-waserver-0.hwa-test TCPADDR 31111
      SECUREADDR 443
      DOMAIN MASTERDM
      FOR MAESTRO
        TYPE MANAGER
        AUTOLINK ON
        BEHINDFIREWALL OFF
        SECURITYLEVEL FORCE_ENABLED
   ```

```
     FULLSTATUS ON
END
```

5. To make the changes effective, run the following command:

   ```
   JnextPlan -for 0000
   ```

6. Optionally, you can deploy a new backup master domain manager on the cloud by performing a scale-up of the components listed in the `values.yaml` file. To perform this operation, set the **waserver.replicaCount** parameter to a value higher than 1. You can now optionally uninstall your on-premises backup master domain manager.
7. To edit the FINAL and FINALPOSTREPORT job streams, type the following command:

   ```
   composer mod js your_xa#final@ full
   ```

   where:

   your_xa

   is the name of the extended agent workstation installed with the master domain manager.

   Edit the following section:

   ```
   STREAMLOGON old_tws_user
   ```

   as follows:

   ```
   STREAMLOGON wauser
   ```

8. Delete the FINAL and FINALPOSTREPORTS job streams from the plan, as follows:

   ```
   conman "canc your_xa#FINALPOSTREPORTS"
   ```

   ```
   conman "canc your_xa#FINAL"
   ```

9. Submit first the FINAL, and then the FINALPOSTREPORTS job streams into the current plan, as follows:

   ```
   conman sbs your_xa#FINAL
   ```

   ```
   conman sbs your_xa#FINALPOSTREPORTS
   ```

10. Reset the value of the **limit** job stream keyword for the FINAL and FINALPOSTREPORTS job streams, both in the database and in the plan, as follows:

    ```
    conman "limit your_xa#FINAL ;10"
    ```

    ```
    conman "limit your_xa#FINALPOSTREPORTS ;10"
    ```

11. To have your dynamic agents connect to the on-cloud master domain manager, copy the certificates located in `/home/wauser/wadata/ITA/cpa/ita/cert/` and duplicate them to `/datadir/ITA/cpa/ita/cert`.
    Perform this operation for each on-premises dynamic agent in your environment.
    You have now successfully switched your master domain manager to the cloud and set up your dynamic agents to work in SSL mode with the on-cloud master domain manager.

Part VI. Troubleshooting installation, migration, and uninstallation

An overview on troubleshooting installation, migration, and uninstallation of the IBM Workload Scheduler.

Issues dealing with the installation, removal, and configuration of IBM Workload Scheduler and its prerequisites.

For information about issues on the DB2® installation, see the DB2® product documentation.

Installation log files

The type of log files you find on your system depends on the type of installation you performed.

On UNIX operating systems, the storage of data generated by IBM® Workload Scheduler, such as logs and configuration files, are stored by default in the *DATA_DIR* directory, which you can optionally customize at installation time. By default, this directory is `<TWA_home>/TWSDATA` for the server and agent components, and `<DWC_home>/DWC_DATA` for the Dynamic Workload Console. The product binaries are stored instead, in the installation directory. For more information, see Server components installation - serverinst script *(on page 310)*, Dynamic Workload Console installation - dwcinst script *(on page 320)*, and Agent installation parameters - twsinst script *(on page 84)*.

Note: If you deployed the product components using Docker containers, this is the default behavior and it cannot be modified. However, if you installed the product components using the command-line installation, the **--data_dir** parameter can be used to change the path.

master domain manager or dynamic domain manager and its backup

`<TWA_home>/TWSDATA/installation/logs`

Dynamic Workload Console

`<DWC_home>/DWC_DATA/installation/logs`

Dynamic agents and fault-tolerant agents

`<INST_DIR>/TWSDATA/installation/logs/twsinst_<operating_system>_<TWS_user>^<product_version_number>.log`. For more information, see The twsinst log files *(on page 279)*.

On Windows operating systems, installation log files are stored in the following paths:

master domain manager or dynamic domain manager and its backup

`<INSTALL_DIR>\logs`

Dynamic Workload Console

`<INSTALL_DIR>\logs`

When you install a fix pack, the suffix at the end of the file name lists the fix pack number in addition to the General Availability version number, for example:

```
serverinst_<version_number>.0.0<fix_pack_number>.log
```

Chapter 1. The twsinst log files

The twsinst log file name is:

On Windows operating systems:

`<TWS_INST_DIR>\logs\twsinst_operating_system_TWS_user^version_number.log`

Where:

TWS_INST_DIR

The IBM Workload Scheduler installation directory. The default installation directory is `C:\Program Files\IBM\TWA_TWS_user`.

operating_system

The operating system.

TWS_user

The name of the user for which IBM Workload Scheduler was installed, that you supplied during the installation process.

On UNIX operating systems:

`<TWS_INST_DIR>/TWSDATA/installation/logs/twsinst_operating_system_TWS_user^product_version_number.log`

Where:

TWS_INST_DIR

The IBM Workload Scheduler installation directory. The default installation directory is `/opt/IBM/TWA_TWS_user`.

operating_system

The operating system.

TWS_user

The name of the user for which IBM Workload Scheduler was installed, that you supplied during the installation process.

Chapter 2. Analyzing return codes for agent installation, upgrade, restore, and uninstallation

Check how your operation completed by analyzing the return codes that are issued by twsinst.

Return codes that you can receive when you are installing, upgrading, restoring, or uninstalling agents. To analyze them and take corrective actions, run the following steps:

On Windows operating systems

1. Display the operation completion return code, by using the following command:

   ```
   echo %ERRORLEVEL%
   ```

2. Analyze the following table to verify how the operation completed:

Table 22. Windows operating system agent return codes

Error Code	Description	User action
0	Success: The operation completed successfully without any warnings or errors.	None.
1	Generic failure	Check the messages that are displayed on the screen by the script. Correct the error and rerun the operation. If the error persists, search the https://www.ibm.com/support/home/ database for a solution.
2	The installation cannot create the IBM Workload Scheduler user or assign the correct permission to it.	Verify the operating system policies and configuration. Verify the input values. If necessary, create the user manually before you run the installation.
3	The password is not correct or the installation cannot verify it.	Verify the operating system policies and configuration. Verify the input values.
4	The IBM Workload Scheduler installation directory is not empty. You specified as installation folder a directory that exists.	Empty it or specify a different directory.
5	An error occurred checking the IBM Workload Scheduler prerequisites on the workstation.	See the System Requirements Document at IBM Workload Scheduler Detailed System Requirements.
6	The IBM Workload Scheduler registry is corrupted.	Use the recovInstReg option to recover the registry. Then, rerun the operation.
7	The upgrade or restore operation cannot retrieve the information from the configuration files.	Check that the previous installation and the `localopts`, the `globalopts`, the `ita.ini`, and the `JobManager.ini` files are not corrupted. Correct the errors and try again the operation.
8	The upgrade, restore, or uninstallation cannot proceed because there are jobs that are running.	Stop the jobs that are running or wait for these jobs to complete. Restart the operation.
9	The upgrade, restore, or uninstallation cannot proceed because there are files that are locked.	Stop all the processes that are running and close all the activities that can block

Error Code	Description	User action
		the installation path. Restart the operation.
10	The upgrade, restore, or uninstallation cannot proceed because there are command lines opened.	Close the command lines. Restart the operation.

On UNIX and Linux operating systems:

1. Display the installation completion return code, by using the following command:

```
echo $?
```

2. Analyze the following table to verify how the installation completed:

Table 23. UNIX or Linux operating system agent return codes

Error Code	Description	User action
0	Success: The installation completed successfully without any warnings or errors.	None.
1	Generic failure.	Check the messages that are displayed on the video by the script. Correct the error and rerun the operation. If the error persists, search the https://www.ibm.com/support/home/ database for a solution.
2	The installation did not find the IBM Workload Scheduler user or its home directory. The IBM Workload Scheduler user that you specified either does not exist or does not have an associated home directory.	Verify the operating system definition of the IBM Workload Scheduler user.
3	Not applicable	
4	The IBM Workload Scheduler installation directory is not empty. You specified as installation folder a directory that exists.	Empty it or specify a different directory.
5	An error occurred checking the IBM Workload Scheduler prerequisites on the workstation.	See the System Requirements Document at IBM Workload Scheduler Detailed System Requirements.
6	The IBM Workload Scheduler registry is corrupted.	Use the recovInstReg option to recover the registry. Then, rerun the operation.
7	The upgrade or restore operation cannot retrieve the information from the configuration files.	Check that the previous installation and the `localopts`, the `globalopts`, the `ita.ini`, and the `JobManager.ini` files are not corrupted. Correct the errors and try again the operation.
8	The upgrade, restore, or uninstallation cannot proceed because there are jobs that are running.	Stop the jobs that are running or wait for these jobs to complete. Restart the operation.

Error Code	Description	User action
9	The upgrade, restore, or uninstallation cannot proceed because there are files that are locked.	Stop all the processes that are running and close all the activities that can block the installation path. Restart the operation.
10	The upgrade, restore, or uninstallation cannot proceed because there are command lines opened.	Close the command lines. Restart the operation.

Chapter 3. Problem scenarios: install, reinstall, upgrade, migrate, and uninstall

Known problems and troubleshooting

This section describes known problem scenarios that could occur with the installation, re-installation, upgrade, migration, and uninstallation of IBM® Workload Scheduler components.

Installation or upgrade fails on RHEL version 9 and later

Installing or upgrading on RHEL version 9 and later fails if you were using default certificates.

Problem scenario

You are using a product version earlier than 10.2.1 with default certificates and you plan to upgrade to version 10.2.3, or you have upgraded to version 10.2.1 with default certificates and now plan to upgrade to 10.2.3. This problem can also occurs if you perform a parallel upgrade from versions 9.4 or 9.5, which require a fresh installation of IBM® Workload Scheduler components. If one or more IBM® Workload Scheduler components are installed on RHEL version 9 or later, the upgrade or fresh installation fails.

You might encounter an error message similar to the following:

```
AWSRES003E The REST service cannot be contacted. Check if the service is
running or the existence of firewall rules or some issues on the dns side resolving
the server hostname that could prevent contacting the service.
```

Cause and solution

The SHA-1 signatures contained in the IBM® Workload Scheduler default certificates are not supported by the OpenSSL libraries embedded in RHEL version 9 or later. This is a known problem with RHEL version 9 and later. For more information, see Bug 2055796 - Enable SHA-1 signatures through LEGACY policy configuration.

To work around this problem, perform the following steps:

1. Stop all IBM® Workload Scheduler services and WebSphere Application Server Liberty, by running the following commands:

   ```
   conman stop; wait
   conman shut; wait
   conman ShutDownLwa
   stopappserver
   ```

2. Browse to the following paths:

   On UNIX™ operating systems

   TWA_DATA_DIR`\ssl`

   On Windows™ operating systems

   `installation_dir``\TWS\ssl`

3. Edit the `openssl.cnf` file as follows:
   - add the **alg_section = evp_properties** property in section **[openssl_init]**.
   - create a new section named **[evp_properties]** with this content:

     ```
     #to enable in RHEL-9 using the embedded OpenSSL 3.0.x the support of SHA-1
     #for signature creation and verification
     rh-allow-sha1-signatures = yes
     ```

4. Restart all IBM® Workload Scheduler and WebSphere Application Server Liberty services by running the following commands:

   ```
   conman start
   conman startappserver
   ```

```
StartUpLwa
```

Error in testing a connection or running reports on an engine returned from Fix Pack 1 to GA level when using an MSSQL database

If you install General Availability (GA) version 9.5 and a fix pack on a master domain manager using an MSSQL database and then return the workstation to GA version 9.5, you might experience problems when testing the engine connection and running reports.

When you try to test the engine connection or run a report, the operation fails and the following messages are displayed in the Dynamic Workload Console:

- AWSUI0803W Test connection to *engine_name*: engine successful, database failed.
- AWSUI0360E The JDBC URL is not configured on the selected engine, so the reporting capabilities cannot be used. Contact the IBM® Workload Scheduler administrator.

Cause and solution:

The reporting feature for the MSSQL databases is released with version 9.5, Fix Pack 1. If you return the master domain manager to the GA version, you can no longer use the reporting feature for the MSSQL databases. To continue working with the Dynamic Workload Console, disable the database configuration for the reporting feature by performing the following steps:

1. Log in to the Dynamic Workload Console and select Administration > Manage Engines.
2. Click on the engine you returned to the GA version.
3. In the **Database Configuration for Reporting** section, disable the **Enable Reporting** check box.

.

Error in upgrading the IBM® Workload Scheduler database when using a DB2 database

When you run the configureDB script to upgrade DB2 when upgrading to IBM® Workload Scheduler 9.5 or later, the following error messages are returned:

- ALTER TABLE LOG.LLRC_LOG_RECORDS ADD COLUMN LLRC_DIFFERENCE VARCHAR (4095) DB21034E. The command was processed as an SQL statement because it was not a valid Command Line Processor command.
- QL0670N The statement failed because the row or column size of the resulting table would have exceeded the row or column size limit: "8101". Table space name: "LOG_DAT_8K". Resulting row or column size: "10000". SQLSTATE=54010

Cause and solution:

If you try to upgrade IBM® Workload Scheduler to version 9.5 or later, and the IBM® Workload Scheduler database was created with DB2, the DB2 option **EXTENDED_ROW_SZ** remains set to DISABLE during the upgrade process.

Starting from IBM® Workload Scheduler version 9.5, the LOG. LLRC_LOG_RECORDS table exceeds the table space or buffer pool page size which was previously set to 8 kilobytes and this causes the upgrade process to fail.

You can solve the problem by either changing the EXTENDED_ROW_SZ DB2 configuration parameter or, if you do not want to change this parameter, migrate the tables to a new buffer pool and table space with a page size of 16 kilobytes:

Change the DB2 configuration parameter

Change the DB2 configuration parameter EXTENDED_ROW_SZ to ENABLE.

OR

Create a new buffer pool and table space and migrate the tables to the new table space

1. Create a new buffer pool and table space with a page size of 16 kilobytes instead of 8 kilobytes.
2. Migrate the involved tables, which are defined in the LOG schema, to the new table space.

Problems in encrypting the useropts file

You have upgraded from a version earlier than 10.2 with encryption automatically enabled, but the `useropts` file is not encrypted.

To solve this problem, launch the following command on the master domain manager:

```
UpdateUseropts -update twsuser twsuserpassword
```

where:

twsuser

is the name of the user whose password you want to encrypt.

twsuserpassword

is the password you want to encrypt.

The `useropts` file is encrypted immediately.

For more information about the `useropts` file, see Setting user options *(on page)*.

For more information about enabling product encryption after upgrading, see Enabling product encryption after upgrading *(on page 262)*.

WebSphere Application Server Liberty server does not start when applying a fix pack to the backup master domain manager

A failure occurs when applying version 9.5, Fix Pack 4, or later, to a previous fix pack.

If the upgrade process fails starting the Liberty application server, with a message similar to the following:

```
WAINST200I Configuring WLP.

    WAINST015E The following command failed:

    C:\WA\BKM95\appservertools\startAppServer.bat -directclean

    WAINST035I For more details see the installation log file: C:\WA\BKM95\logs\serverinst_9.5.0.04.log.
```

Cause and solution:

It might occur that the previous WebSphere Application Server Liberty process, named **javaw**, is still up and running and is already using the application ports.

To solve the problem, proceed as follows:

1. Check if there is a **javaw** process running which is related to the previous version 9.5 fix pack *x* instance, using the Java version installed in the `JavaExt9.5.0._OLD_FP` path, for example `JavaExt9.5.0.02\jre\jre\bin\javaw.exe`.
2. If you find the **javaw** process, stop it and restart the upgrade process.

Error received when creating MSSQL database

Error received when creating MSSQL database

When creating the database for MSSQL, you might receive an error similar to the following:

```
'CREATE SCHEMA' must be the first statement in a query batch.
```

Cause and solution

When you run the configureDb script specifying the `execsql=false` parameter, the `customSQL.sql` and `customSQLAdmin.sql` are created and stored locally.

Before sending them to the database administrator, perform the following steps:

1. Add the following to strings to the `customSQL.sql` file:

```
CREATE SCHEMA EVT
GO
CREATE SCHEMA PLN
GO
CREATE SCHEMA MDL
GO
CREATE SCHEMA LOG
GO
CREATE SCHEMA DWB
GO
```

2. Replace all semicolons (;) with the string `GO` in the `customSQL.sql`.
3. Send both files to the database administrator.
4. The database administrator must run the `customSQLAdmin.sql` file on the database server.
5. The database administrator must run the `customSQL.sql` file on the new database created with the previous query.

For more information about the `execsql` parameter and the configureDb script, see Database configuration - configureDb script *(on page 301)*.

Chapter 4. Uninstalling IBM Workload Scheduler manually

Steps to take when manually uninstalling the IBM Workload Scheduler master domain manager.

How to manually remove the IBM Workload Scheduler master domain manager.

Run the steps listed in the following topics to manually uninstall an IBM Workload Scheduler instance:

- Uninstalling manually on Windows operating systems *(on page 287)*
- Uninstalling manually on UNIX operating systems *(on page 288)*

Read the following topic to learn about known workaround for problems that might affect the IBM Workload Scheduler uninstall:

- Problems during manual uninstall *(on page 290)*

Uninstalling manually on Windows™ operating systems

Steps to take when manually uninstalling the IBM Workload Scheduler master domain manager on a Windows™ operating systems.

Run the following steps to manually remove an IBM Workload Scheduler master domain manager.

Note: If your RDBMS is based on Oracle, browse to the *TWA_home*`\usr\servers\engineServer\configDropins\overrides` path and check in the `datasource.xml` configuration file the net service name used for your database before uninstalling the master domain manager.

1. Shut down all IBM Workload Scheduler operations and processes

1. On a system prompt, go to the IBM Workload Scheduler installation path.
2. Set the environment by running the `twa_env.cmd` command.
3. Stop the dynamic agent by running the `ShutDownLwa` command.
4. Stop **netman**, **conman** and their child processes by running the `conman “shutdown` command.
5. Stop the event process by running the `conman stopmon` command.
6. Stop the application server process by running the `conman stopappservman` command.
7. In the task manager, verify that the following processes are inactive:

```
netman
appservman
java
mailman
monman
```

As an alternative, you can also stop all processes by shutting down the related IBM Workload Scheduler services from the services panel.

2. Delete the IBM Workload Scheduler services

If you are uninstalling the master domain manager, you must delete the following services:

```
tws_tokensrv_TWS_user
tws_maestro_TWS_user
tws_ssm_agent_TWS_user
tws_netman_TWS_user
tws_cpa_agent_TWS_user
IBMWASService - TWS_user
```

The command to delete a service is:

```
sc delete service_name
```

When you finished, check that the following services are no longer listed in the active services for the *TWS_user*:

Workload Scheduler
Netman
Token service
Common Platform agent

If any of these services is still in the list, reboot the system and check again.

3. **Delete the IBM Workload Scheduler files**

   Delete all the files under the `TWA_install_dir` directory.

4. **Drop the IBM Workload Scheduler tables to the RDBMS**

   On DB2:

   Run the following steps:

   1. From the program menu, open the DB2 command line processor (CLP).
   2. Look for the database name by running the command:

      ```
      list db directory
      ```

   3. If you see an entry named *`your_db_name`* associated to the IBM Workload Scheduler instance, run the command:

      ```
      drop db your_db_name
      ```

   If the master domain manager was installed on the DB2 client, run steps 1 and 5 also on the system where the master domain manager is installed.

   On ORACLE:

   Run the following steps:

   1. Access the ORACLE command line.
   2. Run the command:

      ```
      sqlplus system/password@net_service_name
      ```

   3. Delete all the tables related to the IBM Workload Scheduler instance by running the command:

      ```
      drop user ORACLE_TWS_user cascade;
      ```

Uninstalling manually on UNIX™ operating systems

Steps to take when uninstalling IBM Workload Scheduler master domain manager manually on UNIX™ operating systems.

To manually remove an IBM Workload Scheduler master domain manager complete the following steps.

Note: If your RDBMS is based on Oracle, browse to the *TWA_DATA_DIR*`/usr/servers/engineServer/configDropins/overrides` path and check in the `datasource.xml` configuration file the net service name used for your database before uninstalling the master domain manager.

1. **Shut down all IBM Workload Scheduler operations and processes**

   1. On a system prompt, go to the IBM Workload Scheduler installation path.
   2. Set the environment by running the `twa_env.sh` command.
   3. Stop the dynamic agent by running the `ShutDownLwa` command.
   4. Stop the event processor by running the conman stopmon command.
   5. Stop the application server process by running the conman stopappservman command.
   6. Stop **netman**, **conman**, and their child processes by running the `conman "shut;wait"` command.
   7. To verify that the following processes are inactive, run the command `ps -ef | grep process_name`.

      ```
      netman
      appservman
      ```

```
java
mailman
monman
```

2. Delete the IBM Workload Scheduler files

Delete all the files under the `TWS_install_dir` directory.

Note: The `TWS_install_dir` directory is not the IBM Workload Automation directory, as that might also contain a Dynamic Workload Console installation.

3. Drop the IBM Workload Scheduler tables into the RDBMS

On DB2:

Complete the following steps:

1. From the program menu, open the DB2 command-line processor (CLP)
2. Look for the database name by running the command:

   ```
   list db directory
   ```

3. If you see an entry named *your_db_name* associated to the IBM Workload Scheduler instance, run the command:

   ```
   drop db your_db_name
   ```

4. If you see an entry named *your_db_name* associated to the IBM Workload Scheduler instance, run the command:

   ```
   uncatalog db your_db_name_DB
   ```

5. To see which node is attached to the master domain manager, run the command:

   ```
   list node directory
   ```

6. Run the command:

   ```
   uncatalog node your_node
   ```

If the master domain manager was installed on the DB2 client, perform the same procedure also on the workstation where the master domain manager is installed.

On ORACLE:

Complete the following steps:

1. Access the Oracle command line.
2. Run the command:

   ```
   sqlplus system/password@net_service_name
   ```

3. Delete all the tables related to the IBM Workload Scheduler instance by running the command:

   ```
   drop user ORACLE_TWS_user cascade;
   ```

4. Delete the IBM Workload Scheduler administrative user that was created at installation time.

5. Delete the IBM Workload Automation and the IBM Workload Scheduler registries

1. Edit the `/etc/TWS/TWSRegistry.dat` file.
2. Delete the lines tagged with **TWS_user**.
3. Go to the `/etc/TWA` directory which contains two files for each IBM Workload Scheduler instance installed.
4. Look for the properties file that applies to the IBM Workload Scheduler instance to remove.

5. Delete the properties file and the file with the same filename and extension `.ext`.
6. Delete the `/etc/init.d/tebetl-tws_cpa_agent_`*TWS_user* directory.

6. Remove the Common Platforms Agent configuration file

Remove the file named `/etc/teb/teb_tws_cpa_agent_`*`TWS_user`*`.ini`.

7. Remove WebSphere Application Server Liberty

Delete all files located in the *`IWA_install_dir`*`/wlp` directory and the `wlp` directory itself.

Note: Do not delete the above files and directories if other components are installed and using WebSphere Application Server Liberty, such as the Dynamic Workload Console.

Problems during manual uninstall

The following problem might occur during a manual uninstall:

- File deletion on Windows too slow *(on page 290)*

File deletion on Windows™ too slow

When manually deleting files during a manual uninstallation, the deletion of the files in the path `$TWA_DIR\TWS\stdlist\yyyy.mm.dd\Onnnn.hhmm` is unacceptably slow.

Cause and solution:

This problem is caused by a known Microsoft™ issue on Windows™ operating systems. It occurs when you try to delete the indicated files on the Windows™ system after having uninstalled the master domain manager. To prevent the problem from occurring use **Shift-Canc** to remove these files instead of using the **Delete** menu option, moving them to the recycle bin, or using the **Canc** key on the keyboard.

Part VII. Uninstalling

An overview on how to uninstall the product.

Uninstalling the product does not remove files created after IBM Workload Scheduler was installed, nor files that are open at the time of uninstallation. If you do not need these files, you must remove them manually. If you intend to reinstall and therefore need to use the files, make a backup before starting the installation process. The uninstallation does not remove your DB2® or Oracle database.

Note: To manually uninstall IBM Workload Scheduler, see Uninstalling IBM Workload Scheduler manually *(on page 287)*

Chapter 1. Uninstalling the main components

Before performing the uninstallation, whether if the following conditions are met:

1. Ensure that the user running the process has the following authorization requirements:

   Windows™ operating systems

   If you set the Windows User Account Control (UAC), your login account must be a member of the Windows™ **Administrators** group or domain administrators with the right, **Act as Part of the Operating System**.

   If you set the Windows User Account Control (UAC) on the workstation, you must run the installation as **administrator**.

   UNIX™ and Linux™ operating systems

   If the component was installed with root privileges, **root** access is required. If you performed a **no-root installation**, specify the same user used for installing the component.
2. Ensure that all IBM Workload Scheduler processes, services and the WebSphere Application Server Liberty process are stopped, and that there are no active or pending jobs. For information about stopping the processes and services see Starting and stopping processes on a workstation *(on page)*.

The following section describes how to uninstall the following components:

- master domain manager or its backup
- dynamic domain manager or its backup
- agents

The uninstallation removes the product files, the registry keys, and on Windows operating systems, also the services. It also removes the binaries related to the installed IBM Workload Scheduler agent.

The uninstallation program does not remove the IBM Workload Scheduler configuration files.

Uninstalling a backup master domain manager

To uninstall a backup master domain manager, perform the following steps:

1. To uninstall the backup master domain manager, you must first remove it from the plan. Set the workstation running the backup master domain manager to `ignore`, using either the `composer mod cpu workstation_name>` command or from the Dynamic Workload Console.
2. Run JnextPlan to generate the new production plan so that the backup master domain manager is removed from the plan.
3. Run the uninstall script.

   a. Change directory using the following command:

   ```
   cd TWA_home>/TWS/tws_tools
   ```

   b. Run the uninstallation process by running the script as follows:

   Windows™ operating systems

   ```
   cscript uninstall.vbs --prompt no --wauser user_name>
   ```

   UNIX™ and Linux™ operating systems

   ```
   ./uninstall.sh --prompt no --wauser user_name
   ```

   where, *user_name>* represents the user for which you want to uninstall the backup master domain manager. The procedure runs without prompting the user to confirm the uninstallation.
4. Run JnextPlan to update the plan with the changes.

Uninstalling a master domain manager

To uninstall a master domain manager, perform the following steps:

1. Run the uninstall script.
    a. Change directory using the following command:
    ```
    cd TWS_home/TWS/tws_tools
    ```
    b. Start the uninstallation process by running the script as follows:
    Windows™ operating systems
    ```
    cscript uninstall.vbs --prompt no --wauser user_name
    ```
    UNIX® and Linux® operating systems
    ```
    ./uninstall.sh --prompt no --wauser user_name
    ```
    where, *user_name* represents the user for which you want to uninstall the master domain manager. The procedure runs without prompting the user to confirm the uninstallation.
2. Drop the IBM Workload Scheduler tables to the RDBMS.
    On DB2®:
    Run the following steps:
    a. From the program menu, open the DB2® command-line processor (CLP).
    b. Look for the database name by running the command:
    ```
    list db directory
    ```
    c. If you see an entry named *your_db_name* associated to the IBM Workload Scheduler instance, run the command:
    ```
    drop db your_db_name
    ```
    d. If you see an entry named *your_db_name*_DB associated to the IBM Workload Scheduler instance, run the command:
    ```
    uncatalog db your_db_name_DB
    ```
    e. To see which node is attached to the master domain manager system run the command:
    ```
    list node directory
    ```
    f. Run the command:
    ```
    uncatalog node your_node
    ```
    If the master domain manager was installed on the DB2® client, run the same on the system where the master domain manager is installed.
    On ORACLE:
    Run the following steps:
    a. Access the ORACLE command line.
    b. Run the command:
    ```
    sqlplus system/password@net_service_name
    ```
    c. Delete all the tables related to the IBM Workload Scheduler instance by running the command:
    ```
    drop user ORACLE_TWS_user cascade;
    ```
3. Delete the IBM Workload Scheduler administrative user that was created at install time.

The log files generated from this command are located in the following path:

On Windows operating systems

TWA_home\logs

On UNIX operating systems

TWA_DATA_DIR/installation/logs

Uninstalling the Dynamic Workload Console

Ensure that all IBM Workload Scheduler processes, services and the WebSphere Application Server Liberty process are stopped, and that there are no active or pending jobs. For information about stopping the processes and services see Starting and stopping processes on a workstation *(on page)*.

To uninstall the Dynamic Workload Console, perform the following steps:

1. Change directory to the folder containing the uninstallation script:

   ```
   cd DWC_INST_DIR/tools
   ```

2. Run the uninstallation process by running the script as follows:

   Windows™ operating systems

   ```
   cscript uninstall.vbs --prompt no
   ```

   UNIX™ and Linux™ operating systems

   ```
   ./uninstall.sh --prompt no
   ```

   The procedure runs without prompting the user to confirm the uninstallation.

The log file generated by this command are located in:

On Windows operating systems

<*DWC_home*>\logs

On UNIX operating systems

<*DWC_DATA_dir*>/installation/logs

Uninstalling a dynamic domain manager or its backup

Authorization requirements to verify before uninstalling.

1. Ensure that all IBM Workload Scheduler processes, services and the WebSphere Application Server Liberty process are stopped, and that there are no active or pending jobs. For information about stopping the processes and services, Starting and stopping processes on a workstation *(on page)*.
2. To maintain a correct hierarchy of the IBM Workload Scheduler network, see Uninstalling a dynamic domain manager maintaining a correct hierarchy in the network *(on page 295)*.

To uninstall a dynamic domain manager or its backup, perform the following steps:

1. Run the uninstall script.

   a. Change directory using the following command:

   ```
   cd <TWS_home>/TWS/tws_tools
   ```

   b. Start the uninstallation process by running the script as follows:

   Windows™ operating systems

   ```
   cscript uninstall.vbs --prompt no --wauser user_name>
   ```

   UNIX® and Linux® operating systems

```
./uninstall.sh --prompt no --wauser user_name>
```

where, *user_name>* represents the user for which you want to uninstall the dynamic domain manager. The procedure runs without prompting the user to confirm the uninstallation.

2. Drop the IBM Workload Scheduler tables to the RDBMS.

   On DB2®:

   Run the following steps:
   a. From the program menu, open the DB2® command-line processor (CLP).
   b. Look for the database name by running the command:

   ```
   list db directory
   ```

   c. If you see an entry named *your_db_name* associated to the IBM Workload Scheduler instance, run the command:

   ```
   drop db your_db_name
   ```

   d. If you see an entry named *your_db_name*_DB associated to the IBM Workload Scheduler instance, run the command:

   ```
   uncatalog db your_db_name_DB
   ```

   e. To see which node is attached to the dynamic domain manager system run the command:

   ```
   list node directory
   ```

   f. Run the command:

   ```
   uncatalog node your_node
   ```

   If the dynamic domain manager was installed on the DB2® client, run the same on the system where the dynamic domain manager is installed.

   On ORACLE:

   Run the following steps:
   a. Access the ORACLE command line.
   b. Run the command:

   ```
   sqlplus system/password@net_service_name
   ```

   c. Delete all the tables related to the IBM Workload Scheduler instance by running the command:

   ```
   drop user ORACLE_TWS_user cascade;
   ```

3. Delete the IBM Workload Scheduler administrative user that was created at install time.

Uninstalling a dynamic domain manager maintaining a correct hierarchy in the network

To correctly uninstall a dynamic domain manager, perform the following steps:

1. Uninstall the dynamic agents connected to the dynamic domain manager you want to uninstall by using one of the procedures described in this section.
2. In the database, delete the definitions of the workstations of type AGENT that are connected to the dynamic domain manager that you are uninstalling. You can use either the Dynamic Workload Console workload designer or run the following command:

   ```
   composer del ws agent_workstation_name
   ```

3. Delete the definitions of the workstations of type REM-ENG connected to the dynamic domain manager that you are uninstalling. You can use either the Dynamic Workload Console workload designer or run the following command:

   ```
   composer del ws rem_eng_workstation_name
   ```

4. Delete the definitions of the workstations of type POOL connected to the dynamic domain manager that you are uninstalling. You can use either the Dynamic Workload Console workload designer or run the following command:

```
composer del ws pool_workstation_name
```

5. Delete the definitions of the workstations of type D-POOL connected to the dynamic domain manager that you are uninstalling. You can use either the Dynamic Workload Console workload designer or run the following command:

   ```
   composer del ws dpool_workstation_name
   ```

6. Uninstall the dynamic domain manager.
7. Delete the definition of the workstations of type X-AGENT hosted by the dynamic domain manager that you are uninstalling. You can use either the Dynamic Workload Console workload designer, or run the following command:

   ```
   composer del ws x-agent_workstation_name
   ```

8. Delete the definitions of the workstations of type BROKER of the dynamic domain manager that you are uninstalling. You can use either the Dynamic Workload Console workload designer or run the following command:

   ```
   composer del ws broker_workstation_name
   ```

Uninstalling agents using the twsinst script

Before you begin

1. Before starting to uninstall, verify that the user running the uninstallation process has the following authorization requirements:

   Windows™ operating systems

   If you set the Windows User Account Control (UAC), your login account must be a member of the Windows™ **Administrators** group or domain administrators with the right, **Act as Part of the Operating System**.

   If you set the Windows User Account Control (UAC) on the workstation, you must run the installation as **administrator**.

   On UNIX™ and Linux™ operating systems:

   To uninstall a fault-tolerant agent or a dynamic agent that was installed by the **root** user, the user must have **root** access.

   To uninstall a fault-tolerant agent or a dynamic agent that was installed by a **non-root user**, the uninstaller must use the same login used to install the agent. To find the login value used at installation time for dynamic agents, see the read-only `InstallationLoginUser` parameter in the `JobManager.ini` configuration file on the agent.

2. Ensure that you have enough temporary space before starting the uninstallation process.
3. Ensure that all IBM Workload Scheduler processes and services are stopped, and that there are no active or pending jobs. For information about stopping the processes and services, see Starting and stopping processes on a workstation *(on page)*.

Follow these steps to uninstall IBM Workload Scheduler agents using the twsinst script. Depending on the operating system, proceed as follows:

On Windows™ operating systems:

1. Ensure that all IBM Workload Scheduler processes and services are stopped, and that there are no active or pending jobs. For information about stopping the processes and services, see Starting and stopping processes on a workstation *(on page)*.
2. Log on as administrator on the workstation where you want to uninstall the product.
3. **twsinst** for Windows™ is a Visual Basic Script (VBS) that you can run in CScript and WScript mode, from the *installation_dir*\TWS, run the twsinst script as follows:

   ```
   cscript twsinst -uninst -uname username [-wait minutes]
   [-lang lang_id]
   [-work_dir working_dir]
   ```

The uninstallation is performed in the language of the locale and not the language set during the installation phase. If you want to uninstall agents in a language other than the locale of the computer, run the **twsinst** script from the *installation_dir*\TWS as follows:

```
cscript twsinst -uninst -uname user_name -lang language
```

where *language* is the language set during the uninstallation.

On UNIX™ and Linux™ operating systems:

1. Log on as root, or as the user who installed the agent, and change your directory to /*installation_dir*/TWS .
2. From the TWS directory, run the twsinst script as follows:

```
twsinst -uninst -uname username [-wait minutes]
[-lang lang_id] [-work_dir working_dir]
```

The uninstallation is performed in the language of the locale and not the language set during the installation phase. If you want to uninstall agents in a language other than the locale of the computer, run the **twsinst** script from the /*installation_dir*/TWS as follows:

```
./twsinst -uninst -uname user_name -lang language
```

where *language* is the language set during the uninstallation.

-uninst

Uninstalls the IBM Workload Scheduler agent.

-uname *username*

The name of the user for which the IBM Workload Scheduler agent is uninstalled. If you installed the agent as the **root user**, this user name is not to be confused with the user performing the uninstallation. If you installed the agent as a **user different form root**, specify the same user name you used at installation time. In this case, the user performing the uninstallation and the user for which the agent is uninstalled are the same.

-wait *minutes*

The number of minutes that the product waits for jobs that are runn-ing to complete before starting the uninstallation. If the jobs do not complete during this interval, the uninstallation stops and an error message is displayed. Valid values are integers or **-1** for the product to wait indefinitely. The default is **60** minutes.

-lang *lang_id*

The language in which the `twsinst` messages are displayed. If not specified, the system LANG is used. If the related catalog is missing, the default C language catalog is used.

Note: The **-lang** option is not to be confused with the IBM Workload Scheduler supported language packs.

-work_dir *working_dir*

The temporary directory used for the IBM Workload Scheduler installation process files deployment.

On Windows™ operating systems:

If you specify a path that contains blanks, enclose it in double quotation marks. If you do not manually specify a path, the path is set to `%temp%\TWA\tws`*`version_number`*`>`, where `%temp%` is the temporary directory of the operating system.

On UNIX™ and Linux™ operating systems:

The path cannot contain blanks. If you do not manually specify a path, the path is set to `/tmp/TWA/tws`*`version_number`*`>`.

The following is an example of a twsinst script that uninstalls the IBM Workload Scheduler agent, originally installed for user named **twsuser**:

On Windows™ operating systems:

```
cscript twsinst -uninst -uname TWS_user
```

On UNIX™ and Linux™ operating systems:

```
./twsinst -uninst -uname TWS_user
```

Uninstalling agents on IBM i systems

How to uninstall dynamic and z-centric agents on IBM i systems.

To uninstall the agents on an IBM i system by using the twsinst script, perform the following steps:

1. Ensure that all IBM Workload Scheduler processes and services are stopped, and that there are no active or pending jobs. For information about stopping the processes and services, see Application server - starting and stopping *(on page)*.
2. Sign on as the user who performed the installation, either **QSECOFR** or an existing user with ALLOBJ authority. If you installed with a user different from **QSECOFR**, use the same user who performed the installation and specify the **allObjAuth** parameter to indicate that the user has the ALLOBJ authority. For more information about this parameter, see Agent installation parameters on IBM i systems *(on page 112)*. You can find the name of the profile used to perform the installation in the `instUser` located in the *`agent_data_dir`*`/installation/instInfo`.
3. **Note:** Only for dynamic agents, you have the option of installing using a user different from **QSECOFR** and with no specific authorizations. In this case, specify the same user who performed the installation.
4. Change your directory to /*installation_dir*/TWS. For example: `/home/user1/TWS` where user1 is the name of IBM Workload Scheduler user.
5. From the `Installation directory\TWS` directory, run the twsinst script as follows:

```
twsinst -uninst -allObjAuth -uname username
[-wait minutes][-lang lang_id] [-work_dir working_dir]
```

-uninst

Uninstalls IBM Workload Scheduler.

uname *username*

The name of the user for which IBM Workload Scheduler is uninstalled. This user name is not the same as the user performing the installation.

-allObjAuth

If you are installing, upgrading, or uninstalling with a user different from the default **QSECOFR** user, this parameter specifies that the user has the required ALLOBJ authority. Ensure the user is existing and has ALLOBJ authority because the product does not verify that the correct authority is assigned. The same user must be specified when installing, upgrading or uninstalling the agent. If you are using the **QSECOFR** user, this parameter does not apply.

-uname *username*

The name of the user for which IBM Workload Scheduler is uninstalled.

If you are using the **QSECOFR** user or a user with **ALLOBJ authority**, this user name is not the same as the user performing the installation. If you are using a user **different from QSECOFR**, the user performing the installation and the user for which the agent is installed are the same.

-wait *minutes*

The number of minutes that the product waits for jobs that are running to complete before starting the uninstallation. If the jobs do not complete during this intervals the uninstallation stops and an error message is displayed. Valid values are integers or **-1** for the product to wait indefinitely. The default is **60** minutes.

-lang *lang_id*

The language in which the `twsinst` messages are displayed. If not specified, the system LANG is used. If the related catalog is missing, the default C language catalog is used.

-work_dir ***working_dir***

The temporary directory used for the IBM Workload Scheduler installation process files deployment. If you do not manually specify a path, the path is set to `/tmp/TWA/tws`*`version_number`*`>`.

The following example shows a twsinst script that uninstalls the IBM Workload Scheduler agent, originally installed for **twsuser** user:

On IBM i systems:

```
./twsinst -uninst -uname TWS_user -allObjAuth
```

The twsinst script log files on IBM i systems

The twsinst log file name is:

Where: `<`*`TWS_INST_DIR`*`>/twsinst_IBM_i_`*`TWS_user^product_version`*`.log`

TWS_INST_DIR

The IBM Workload Scheduler installation directory. The default installation directory is `/home/TWS_user`.

TWS_user

The name of the user for which IBM Workload Scheduler was installed, that you supplied during the installation process.

product_version

Represents the product version. For example, for version 10.2.3 of the product, the value is 10.2.3.00

Reference

Contains the detailed syntax and explanation for all parameters of the commands required for the command-line installation:

.

Optional password encryption - secure script

Optionally encrypt the passwords you use to install, upgrade, and manage IBM® Workload Scheduler.

This section lists and describes the parameters of the secure script. The secure command uses the AES method and prints the encrypted password to the screen or saves it to a file.

Note: Use this script only to encrypt passwords used during the installation and upgrade processes.

You can either:

- Define a custom passphrase by using the **passphrase** argument and defining the **SECUREWRAP_PASSPHRASE** environment variable in the same shell session in which you run the command using the encrypted password. Ensure you set the **SECUREWRAP_PASSPHRASE** environment variable to the same value as the **passphrase** argument. You can use encrypted passwords only in association with the specific passphrase used to encrypt them.
- Use the standard encryption method provided with the secure command. In this case, you simply specify the **password** parameter.

Note: It is important you understand the limits to the protection that this method provides. The custom passphrase you use to encrypt the passwords is stored in clear format in the `passphrase_variables.xml` file, stored in `configureDropin`. To fully understand the implications of this method, it is recommended you read the information provided by WebSphere Application Server Liberty Base at the link Liberty: The limits to protection through password encryption.

Syntax

Windows operating systems:

```
secure {-password password | -in file}[-passphrase passphrase]
[-base64 e] [-out file]
```

UNIX operating systems:

```
./secure {-password password | -in file}[-passphrase passphrase]
[-base64 e] [-out file]
```

z/OS operating systems:

```
./secure {-password password | -in file}[-passphrase passphrase]
[-base64 e] [-out file]
```

Arguments

-password

Specifies the password to be encrypted. This parameter is mutually exclusive with the **-in** parameter.

-in

Specifies the name and path of the file where you have stored the password to be encrypted. This parameter is mutually exclusive with the **-password** parameter.

-passphrase

Optional. Specifies the custom passphrase that is used to generate the key with which the command encrypts the password. If you set this parameter, inform the user who installs IBM Workload Automation that they must define the **SECUREWRAP_PASSPHRASE** environment variable in the same shell from which they run the installation command, and set it to the same value as the **passphrase** argument. On Windows operating systems, the passphrase must be at least 8 characters long.

-base64 e

Specifies that the encoding process uses the **base64** format.

-out

Specifies the path and name of a file where the command stores the encrypted password. If you do not specify this parameter, the encrypted password is printed to the screen.

Examples

To encrypt password `MyPassword` with a strong passphrase, run the following command:

```
secure -password MyPassword -passphrase de85pU!Mb5G2xewPgdVa
```

To encrypt the password stored in file `MyFile` using the default passphrase and save the encrypted password to file `OutputFile`, run the following command:

```
secure -in C:\info\MyFile -out C:\info\OutputFile
```

Database configuration - configureDb script

This script creates and populates the IBM Workload Scheduler database

This script is typically used by the database administrator for creating and populating the IBM Workload Scheduler database. For a typical scenario, see Creating and populating the database *(on page 40)*.

This section lists and describes the parameters that you can use to create and populate the IBM Workload Scheduler database.

You can specify values in the properties file, type them in the command line, or use both methods. If a parameter is specified both in the properties file and in the command line, the command line value takes precedence.

The log files generated from this command are located in the following path:

On Windows operating systems

TWA_home`\logs`

On UNIX operating systems

TWA_DATA_DIR`/installation/logs`

On z/OS operating system

TWA_DATA_DIR`/installation/logs`

Syntax for Windows operating systems

Show command usage

```
configureDb -? | --usage | --help
```

Retrieve the command parameters and values from a file

```
configureDb --propfile | -f  [property_file]
```

General information

```
      [--lang lang_id]
      [--work_dir working_directory]
      [--wlpdir wlp_directory]
      [--componenttype MDM | DDM | DWC ]
      [--dbadminuser db_admin_user]
      --dbadminuserpw db_admin_password
      --rdbmstype|-r DB2 | DB2Z | ORACLE | MSSQL | POSTGRESQL
      [--dbname db_name]
      [--dbuser db_user]
      [--dbport db_port]
      --dbhostname db_hostname
      [--dbdriverpath db_driver_path]
      --auth_type aythentication_type ]
      [--iwstsname table_space_name]
      [--iwstspath table_space_path]
      [--iwslogtsname log_table_space]
      [--iwslogtspath log_path_table_space]
      [--iwsplantsname plan_table_space]
      [--iwsplantspath plan_path_table_space]
      [--execsql execute_sql]
```

Oracle-only configuration options

```
      --dbpassword db_password
      [--usePartitioning true | false ]
      [--Usage_TsTempName IWS_temp_path]
      [--skipdbcheck true | false]
```

DB2 for z/OS-only configuration options

```
      [--zlocationname zOS_location_containing_db]
      [--zbufferpoolname buffer_pool_in_zOS_location]
```

Syntax for UNIX operating systems

Show command usage

```
configureDb -? | --usage | --help
```

Retrieve the command parameters and values from a file

```
configureDb --propfile | -f  [property_file]
```

General information

```
      [--lang lang_id]
      [--work_dir working_directory]
      [--wlpdir wlp_directory]
      [--componenttype MDM | DDM | DWC ]
      [--dbadminuser db_admin_user]
      --dbadminuserpw db_admin_password
      --rdbmstype|-r DB2 | DB2Z | ORACLE | MSSQL  | POSTGRESQL
      [--dbname db_name]
      [--dbuser db_user]
```

```
      [--dbport db_port]
      --dbhostname db_hostname
      [--dbdriverpath db_driver_path]
      [--iwstsname table_space_name]
      [--iwstspath table_space_path]
      [--iwslogtsname log_table_space]
      [--iwslogtspath log_path_table_space]
      [--iwsplantsname plan_table_space]
      [--iwsplantspath plan_path_table_space]
      [--execsql execute_sql ]
```

Oracle-only configuration options

```
      --dbpassword db_password
      [--usePartitioning true | false ]
      [--Usage_TsTempName IWS_temp_path]
      [--skipdbcheck true | false]
```

DB2- and PostgreSQL-only security options

```
   [--sslkeysfolder  keystore_truststore_folder]
   [--sslpassword ssl_password]
   [--dbsslconnection true | false]
```

DB2 for z/OS-only configuration options

```
      [--zlocationname zOS_location_containing_db]
      [--zbufferpoolname buffer_pool_in_zOS_location]
```

Syntax for z/OS operating system

Show command usage

```
configureDb -? | --usage | --help
```

Retrieve the command parameters and values from a file

```
configureDb --propfile | -f  [properties_file]
```

General information

```
      [--lang lang_id]
      [--work_dir working_directory]
      [--wlpdir wlp_directory]
      [--dbadminuser db_admin_user]
      [--componenttype DWC ]
      --dbadminuserpw db_admin_password
      --rdbmstype|-r DB2 | DB2Z | ORACLE | MSSQL | POSTGRESQL
      [--dbname db_name]
      [--dbuser db_user]
      [--dbport db_port]
      --dbhostname db_hostname
      [--dbdriverpath db_driver_path]
      [--iwstsname table_space_name]
      [--iwstspath table_space_path]
      [--iwslogtsname log_table_space]
      [--iwslogtspath log_path_table_space]
      [--iwsplantsname plan_table_space]
      [--iwsplantspath plan_path_table_space]
      [--execsql execute_sql ]
```

DB2 for z/OS-only configuration options

```
      [--zlocationname zOS_location_containing_db]
      [--zbufferpoolname buffer_pool_in_zOS_location]
```

Database configuration parameters

-? | --usage | --help

Displays the command usage and exits.

--propfile|-f [*properties_file*]

Optionally specify a properties file containing custom values for `configureDb` parameters. The default file for the server components is *`image_location`*`/TWS/`*`interp_name`*`/configureDb.properties`, while the default file for the Dynamic Workload Console is *`image_location`*`/configureDb.properties`. Specifying a properties file is suggested if you have a high number of parameters which require custom values. You can also reuse the file with minimal modification for several installations. If you create a custom properties file, specify its name and path with the **-f** parameter.

--lang *lang_id*

The language in which the messages returned by the command are displayed. If not specified, the system LANG is used. If the related catalog is missing, the default C language catalog is used. If neither **--lang** nor LANG are used, the default codepage is set to SBCS. For a list of valid values for these variables, see the following table:

Table 24. Valid values for -lang and LANG parameter

Language	Value
Brazilian Portuguese	pt_BR
Chinese (traditional and simplified)	zh_CN, zh_TW
English	en
French	fr
German	de
Italian	it
Japanese	ja
Korean	ko
Russian	ru
Spanish	es

Note: This is the language in which the installation log is recorded and not the language of the installed component instance. The command installs all languages as default.

--work_dir

The working directory where you extract the installation image. It also contains the output produced by the command, such as the SQL statements if you set the **execsql** parameter to **false**. The default value is `/tmp` on UNIX operating systems and `C:\tmp` on Windows operating systems.

[--wlpdir *wlp_directory*]

The path to WebSphere Application Server Liberty Base installation directory. WebSphere Application Server Liberty Base is used to decrypt the passwords you provide in encrypted form. This parameter is required only if you encrypt your passwords with the {**xor**} or {**aes**} encoding.

--componenttype MDM | DDM | DWC

The IBM® Workload Scheduler component for which the database is installed. This parameter is optional. Supported values are:

MDM

master domain manager. Applies only to distributed operating systems.

DDM

dynamic domain manager. Applies only to distributed operating systems.

DWC

Dynamic Workload Console (it comprises the Federator).

--dbadminuser ***db_admin_user***

The database administrator user who creates the IBM® Workload Scheduler or Dynamic Workload Console schema objects on the database server. This parameter is optional. Depending on the database vendor, the default values are as follows:

db2admin

when **--rdbmstype** is set to `DB2`

sysadm

when **--rdbmstype** is set to `DB2Z`

system

when **--rdbmstype** is set to `ORACLE`

sa

when **--rdbmstype** is set to `MSSQL`

--dbadminuserpw ***db_admin_password***

The password for the DB administrator user who creates the IBM® Workload Scheduler schema objects on the database server. This parameter is required. Special characters are not supported. You can optionally encrypt the password. For more information, see Encrypting passwords (optional) *(on page 39)*.

--rdbmstype|-r ***rdbms_type***

The database type. Supported databases are:

- **DB2**
- **ORACLE**
- **MSSQL** This value applies to MSSQL and supported MSSQL cloud-based databases.
- **POSTGRESQL**
-

This parameter is required and has no default value.

--dbname ***db_name***

The name of the IBM® Workload Scheduler or Dynamic Workload Console database. This parameter is optional and case sensitive. Depending on the component that you are installing and the database vendor, the default values are as follows:

When installing the server components

TWS

when **--rdbmstype** is set to `DB2`

orcl

when **--rdbmstype** is set to `ORACLE`

TWS

when **--rdbmstype** is set to `MSSQL`

null

when **--rdbmstype** is set to `DB2z`

TWS

when **--rdbmstype** is set to `POSTGRESQL`

When installing the Dynamic Workload Console

This parameter is optional and case sensitive. Depending on the component that you are installing and the database vendor, the default values are as follows:

TDWC

when **--rdbmstype** is set to `DB2`

TDWC

when **--rdbmstype** is set to `DB2Z`

orcl

when **--rdbmstype** is set to `ORACLE`

TDWC

when **--rdbmstype** is set to `MSSQL`

TDWC

when **--rdbmstype** is set to `POSTGRESQL`

--dbuser ***db_user***

The database user that has been granted access to the IBM® Workload Scheduler or Dynamic Workload Console tables on the database server. This parameter is optional. Depending on the component that you are installing and the database vendor, the default values are as follows:

When installing the server components

db2tws

when **--rdbmstype** is set to `DB2`

twsora

when **--rdbmstype** is set to `ORACLE`

sa

when **--rdbmstype** is set to `MSSQL`

null

when **--rdbmstype** is set to `DB2Z`

postgres

when **--rdbmstype** is set to `POSTGRESQL`

When installing the Dynamic Workload Console

db2dwc

when **--rdbmstype** is set to `DB2`

root

when **--rdbmstype** is set to `DB2Z`

twsora

when **--rdbmstype** is set to `ORACLE`

sa

when **--rdbmstype** is set to `MSSQL`

--dbport ***db_port***

The port of the database server. This parameter is optional. Depending on the database vendor, the default values are as follows:

50000

when **--rdbmstype** is set to `DB2`

446

when **--rdbmstype** is set to `DB2Z`

1521

when **--rdbmstype** is set to `ORACLE`

1433

when **--rdbmstype** is set to `MSSQL`

5432

when **--rdbmstype** is set to `POSTGRESQL`

--dbhostname *db_hostname*

The host name or IP address of database server. This parameter is required.

--dbdriverpath *db_driver_path*

The path where the database drivers are stored. This parameter is optional. By default, the configuration script references the JDBC drivers supplied with the product images. If your database server is not compatible with the supplied drivers, then contact your database administrator for the correct version to use with your database server and specify the driver path using this parameter. Ensure you provide the same path in the configureDb, serverinst, and dwcinst commands.

--iwstsname|-tn *table_space_name*

The name of the tablespace for IBM® Workload Scheduler or Dynamic Workload Console data. This parameter is optional for all databases with the exception of the Oracle database. The default value for all databases other than Oracle is:

For all operating systems, except z/OS

TWS_DATA

For z/OS operating system

TWSDATA

--iwstspath|-tp *table_space*

The path of the tablespace for IBM® Workload Scheduler or Dynamic Workload Console data. This parameter is optional. The default value for all databases other than Oracle is:

For all operating systems, except z/OS

TWS_DATA

For z/OS operating system

TWSDATA

Only on Windows systems hosting an MSSQL database, ensure the folder for the tablespace is already existing before running the configureDb command and specify the path using this parameter. Specify the path using forward slashes (/), for example: `c:/<my_path>/TWS_DATA`.

--iwslogtsname|-ln *log_table_space*

The name of the tablespace for IBM® Workload Scheduler log. This parameter is optional for all databases with the exception of the Oracle database. The default value for all databases other than Oracle is **TWS_LOG**. This parameter applies only to the server components.

--iwslogtspath|-lp *log_path_table_space*

The path of the tablespace for IBM® Workload Scheduler log. This parameter is optional. The default value for all databases other than Oracle is **TWS_LOG**. This parameter applies only to the server components. Only on Windows systems hosting an MSSQL database, ensure the folder for the tablespace is already existing before running the configureDb command and specify the path using this parameter. Specify the path using forward slashes (/), for example: `c:/<my_path>/TWS_LOG`.

--iwsplantsname|-pn *plan_table_space*

The name of the tablespace for IBM® Workload Scheduler plan. This parameter is optional for all databases with the exception of the Oracle database. The default value for all databases other than Oracle is **TWS_PLAN**. This parameter applies only to the server components.

--iwsplantspath|-pp *plan_path_table_space*

The path of the tablespace for IBM® Workload Scheduler plan. This parameter is optional. The default value for all databases other than Oracle is **TWS_PLAN**. This parameter applies only to the server components.

Only on Windows systems hosting an MSSQL database, ensure that the folder for the tablespace is already existing before running the configureDb command and specify the path using this parameter. Specify the path using forward slashes (/), for example: `c:/<my_path>/TWS_PLAN`.

--execsql|-es ***execute_sql***

Set to **true** to generate and run the SQL file, set to **false** to generate the SQL statement without running it. The resulting files are stored in the path defined in the **--work_dir** parameter. This option is useful if you wan to review the file before running it. This parameter is optional. The default value is **true**.

--auth_type

This parameter applies only to Windows operating systems. Specify the authentication type. Supported values are as follows:

SQLSERVER

Enables MSSQL authentication type. Only the user specified with the **--dbadminuser** parameter has the grants to administer the IBM® Workload Scheduler or Dynamic Workload Console database.

WINDOWS

Enables Windows authentication type. The Windows user you used to log on to the workstation is assigned the grants to administer the IBM® Workload Scheduler or Dynamic Workload Console database.

The default value is **SQLSERVER**.

Oracle-only configuration syntax

--dbpassword ***db_password***

The password for the user that has been granted access to the IBM® Workload Scheduler or Dynamic Workload Console tables on the database server. This parameter is required only if you are using an Oracle database. Special characters are not supported. You can optionally encrypt the password. For more information, see Encrypting passwords (optional) *(on page 39)*.

--usePartitioning

Only applies when installing the master domain manager. Set to **true** if you want to use the Oracle partitioning feature, otherwise set it to **false**. This parameter is optional. The default value is **true**.

--Usage_TsTempName ***IWS_temp_path***

Only applies when installing the master domain manager. The path of the tablespace for IBM Workload Scheduler temporary directory. This parameter is optional. The default value is **TEMP**.

--skipdbcheck

This parameter specifies whether the check on the existence of the Workload Automation schema for the Oracle user is performed or not. By default, the parameter is set to **false** and a check is performed on the Oracle user. If the user does not exist, the script then proceeds to create the user and the Workload Automation schema.

If you have already created your Oracle user, set this parameter to **true**. As a result, the check is skipped and the schema creation is performed also if the Oracle user is already existing.

This parameter is optional.

DB2- and PostgreSQL-only security options

--sslkeysfolder ***keystore_truststore_folder***

The name and path of the folder containing certificates in PEM format. The installation program automatically processes the keystore and truststore files using the password you specify with the **--sslpassword** parameter. The folder must contain the following files:

- **ca.crt**

  The Certificate Authority (CA) public certificate. Note that if certificates being installed are part of a chain consisting of 3 or more certificates (one Root CA, followed by one or more Intermediate CAs, followed by the end user certificate), then this file must contain the Root CA certificate only. Any Intermediate CA certificates must be stored in the `additionalCAs` subfolder, which therefore becomes a mandatory subfolder. Each Intermediate CA must be stored in the `additionalCAs` subfolder in its own file.

  Note: From V10.2.3, if certificates being installed are part of a chain, the ca.crt can contain also the intermediate CAs. In this case, it must begin with one or more intermediate CA certificates and end with the Root ca.

- **tls.key**

  The private key of the end user certificate for the instance to be installed.

- **tls.crt**

  The public part of the previous key, that is the end user certificate.

For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).

You can optionally create a subfolder to contain one or more `*.crt` files to be added to the server truststore as trusted CA, whose name must be `additionalCAs`. This can be used for example to add to the list of trusted CAs the certificate of the LDAP server or DB2 server. Additionally, you can store here any intermediate CA certificate to be added to the truststore. The subfolder must be named **additionalCAs**. Note that if the end user certificate being installed in the instance is part of a chain consisting of 3 or more certificates (one Root CA, followed by one or more Intermediate CAs, followed by the end user certificate), then the Intermediate CAs certificates must be stored in the `additionalCAs` subfolder, which therefore becomes a mandatory subfolder. Each Intermediate CA must be stored in the `additionalCAs` subfolder in its own file.

For further information about how to generate custom certificates, see Managing certificates using Certman *(on page)*.

This parameter is required if you set the **dbsslconnection** parameter to true.

--sslpassword ***ssl_password***

The password for the custom certificates and the path to the folder containing certificates in PEM format with the **sslkeysfolder** parameter.

For more information, see sslkeysfolder *(on page 316)*.

You can optionally encrypt the password using the secure script. For more information, see Optional password encryption - secure script *(on page 300)*.

This parameter is required if you set the **dbsslconnection** parameter to true.

--dbsslconnection

Specify whether you want to enable SSL connection to the database. Supported values are `true` and `false`. The default value is `false`. If you set this parameter to `true`, the **sslkeysfolder** and **sslpassword** parameters become mandatory.

DB2 for z/OS-only configuration syntax

--zlocationname ***zos_location_containing_db***

The name of an already existing location in the z/OS environment that will contain the new database. The default value is **LOC1**.

--zbufferpoolname ***buffer_pool_in zos_location***

The name of an already existing buffer pool created in the location specified by `-zlocationname`. The default value is BP32K.

Comments

Note: The following parameters are also required when installing the master components and their values must be the same:

- **--rdbmstype**
- **--dbhostname**
- **--dbport**
- **--dbname**
- **--dbuser**

Server components installation - serverinst script

The master domain manager, backup domain manager, dynamic domain manager, backup dynamic domain manager, and installation parameters that can be defined for the serverinst script.

This section lists and describes the parameters that are used when running a serverinst script to install the master domain manager and backup domain manager, dynamic domain manager, and backup dynamic domain manager.

Certificates are now required when installing or upgrading IBM® Workload Scheduler. You can no longer install nor upgrade IBM® Workload Scheduler without securing your environment with certificates. The required certificates are:

- ca.crt
- tls.key
- tls.crt

For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).

You can specify values in the properties file, type them in the command line, or use both methods. If a parameter is specified both in the properties file and in the command line, the command line value takes precedence.

The log files generated from this command are located in the following path:

On Windows operating systems

TWA_home`\logs`

On UNIX operating systems

TWA_DATA_DIR`/installation/logs`

Syntax

On Windows™ operating systems:

Show command usage

```
cscript serverinst.vbs -? | --usage | --help
```

Retrieve the command parameters and values from a properties file

```
cscript serverinst.vbs --propfile|-f  [properties_file]
```

General information

```
cscript serverinst.vbs
--acceptlicense yes|no
```

```
[--lang lang_id]
[--inst_dir install_dir]
[--work_dir working_dir]
[--skipcheckprereq true|false]
[--skipcheckemptydir true|false]
[--skipusercheck true|false]
```

Configuration information for the data source

```
 --rdbmstype|-r DB2 | ORACLE | MSSQL | POSTGRESQL
[--dbname db_name]
[--dbuser db_user]
--dbpassword db_password
[--dbport db_port]
 --dbhostname db_hostname
[--dbdriverpath db_driver_path]
[--dbsslconnection true | false]
```

Security options

```
--sslkeysfolder  keystore_truststore_folder
--sslpassword ssl_password
[--enablefips true | false]
```

User information

```
[--wadomain]
[--wauser wa_user]
[--wapassword wa_password]
```

Configuration information for the application server

```
 --wlpdir|-w wlp_directory
[--httpsport https_port]
[--bootstrapport bootstrap_port]
[--bootsecport bootstrap_sec_port]
[--startserver true | false]
```

Configuration information for dynamic scheduling

```
[--displayname agent_name]
[--jmport port_number]
```

Configuration information for the master domain manager

```
[--componenttype MDM | DDM]
```

Configuration options when --componenttype is MDM

```
[--company company_name]
[--hostname hostname]
[--thiscpu workstation]

[--eifport eif_port]
[--brwksname broker_workstation_name]
[--brnetmanport broker_netman_port]
[--netmanport netman_port_number]
[--netmansslport netman_port_number]
```

Configuration options when --componenttype is DDM

```
[--domain domain_name]
```

```
 --master mdm-domain_name
 --mdmhttpsport mdm_https_port
 --mdmbrokerhostnamemdm_hostname
[--eifport eif_port]
[--brwksname broker_workstation_name]
[--brnetmanport broker_netman_port]
[--netmanport netman_port_number]
[--netmansslport netman_port_number]
[--isforzos yes|no]
```

IBM® Workload Scheduler encryption options

```
[--useencryption true | false]
[--encryptionpassword default]
```

On UNIX® operating systems

Show command usage

```
./serverinst.sh -? | --usage | --help
```

Retrieve the command parameters and values from a properties file

```
./serverinst.sh --propfile|-f  [properties_file]
```

General information

```
 ./serverinst.sh
 --acceptlicense yes|no
[--lang lang_id]
[--inst_dir install_dir]
[--work_dir working_dir]
[--data_dir wa_datadir]
[--skipcheckprereq true|false]
[--skipcheckemptydir true|false]
```

Configuration information for the data source

```
 --rdbmstype|-r DB2 | ORACLE | MSSQL |  POSTGRESQL
 [--dbname db_name]
 [--dbuser db_user]
  --dbpassword db_password
 [--dbport db_port]
  --dbhostname db_hostname
 [--dbdriverpath db_driver_path]
 [--dbsslconnection true | false]
```

Security options

```
--sslkeysfolder  keystore_truststore_folder
--sslpassword ssl_password
[--enablefips true | false]
```

User information

```
 [--wauser wa_user]
 [--wapassword wa_password]
```

Configuration information for the application server

```
 --wlpdir|-w wlp_directory
[--httpsport https_port]
[--bootstrapport bootstrap_port]
[--bootsecport bootstrap_sec_port]
[--startserver true | false]
```

Configuration information for dynamic scheduling

```
[--displayname agent_name]
[--jmport port_number]
```

Configuration information for the master domain manager

```
[--componenttype MDM | DDM]
```

Configuration options when --componenttype is MDM

```
[--company company_name]
[--hostname hostname]
[--thiscpu workstation

[--eifport eif_port]
[--brwksname broker_workstation_name]
[--brnetmanport broker_netman_port]
[--netmanport netman_port_number]
[--netmansslport netman_port_number]
```

Configuration options when --componenttype is DDM

```
[--domain domain_name]
 --master mdm_domain_name
 --mdmhttpsport mdm_https_port
 --mdmbrokerhostname mdm_hostname
 --eifport eif_port]
[--brwksname broker_workstation_name]
[--brnetmanport broker_netman_port]
[--netmanport netman_port_number]
[--netmansslport netman_port_number]
[--isforzos yes|no]
```

IBM® Workload Scheduler encryption options

```
[--useencryption true | false]
[--encryptionpassword default]
```

Arguments

? | --usage | --help

Displays the command usage and exits.

--propfile|-f [*properties_file*]

Optionally specify a properties file containing custom values for serverinst parameters. The default file is

On Windows™ systems

image_dir>\TWS95_WIN_X86_64_SERVER\TWS\WINDOWS_X86_64\serverinst.properties

On UNIX® systems

image_dir>/TWS/*interp*>/serverinst.properties

Specifying a properties file is suggested if you have a high number of parameters which require custom values. You can also reuse the file with minimal modification for several installations. If you create a custom properties file, specify its name and path with the **-f** parameter.

General information

--acceptlicense *yes*|*no*

Specify whether to accept the License Agreement.

--lang *lang_id*

The language in which the messages returned by the command are displayed. If not specified, the system LANG is used. If the related catalog is missing, the default C language catalog is used. If neither **--lang** nor LANG are used, the default codepage is set to SBCS. For a list of valid values for these variables, see the following table:

Table 25. Valid values for -lang and LANG parameter

Language	Value
Brazilian Portuguese	pt_BR
Chinese (traditional and simplified)	zh_CN, zh_TW
English	en
French	fr
German	de
Italian	it
Japanese	ja
Korean	ko
Russian	ru
Spanish	es

Note: This is the language in which the installation log is recorded and not the language of the installed component instance. The command installs all languages as default.

--inst_dir ***installation_dir***

The directory of the IBM Workload Scheduler installation. This parameter is optional. The default value is:

On Windows™ operating systems

```
C:\Program Files\wa
```

On UNIX® operating systems

```
/opt/wa
```

--work_dir ***working_dir***

The temporary directory used by the program to deploy the installation process files. This parameter is optional. The default value is:

On Windows™ operating systems

```
C:\TMP
```

On UNIX® operating systems

```
/tmp/wa<version_number>
```

This parameter can also function as a backup directory during product upgrade with path *WORKING_DIR*`/backup`.

--data_dir ***wa_datadir***

UNIX operating systems only. Specify the path to a directory where you want to store the logs and configuration files produced by IBM Workload Scheduler. This parameter is optional. If you do not specify this parameter, all data files generated by IBM Workload Scheduler are stored in the *TWA_home*`/TWSDATA` directory. This path is called, in the publications, *TWA_DATA_DIR*.

--skipcheckprereq ***true|false***

If you set this parameter to `false`, IBM Workload Scheduler does not scan system prerequisites before starting the installation. This parameter is optional. The default value is `true`. For more information about the prerequisite check, see Scanning system prerequisites for IBM Workload Scheduler *(on page 33)*.

--skipcheckemptydir *true|false*

Set this parameter to `true` to avoid checking whether the installation directory is empty. By default, this parameter is `false`, because starting from version 9.5 the installation directory must be empty. If you set this parameter to `true` and the installation directory is not empty, the installation process might fail.

--skipusercheck *true|false*

If you set this parameter to `true`, IBM Workload Scheduler, performs no checks on the user. This parameter is optional. The default value is `false`. By default, the following checks are performed:

local user

The script checks if the specified user is existing, has the correct access rights, and the password specified with the `wapassword` parameter is correct. If the user does not exist, the script creates it and grants it the correct access rights. If the specified password is incorrect, the script returns an error and the installation process stops.

domain user

The script checks if the specified user is existing, has the correct access rights, and the password specified with the `wapassword` parameter is correct. If the user does not exist, the script cannot create it and the installation process ends in error. If the user exists but does not have the correct access rights, the script assigns it the required rights. If the specified password is incorrect, the script returns an error and the installation process stops.

Configuration information for the data source

The values for these parameters must match the values defined by the database administrator when creating the database. For more information, see Creating and populating the database *(on page 40)* and browse to the topic for the database you are using.

--rdbmstype|-r *rdbms_type*

The database type. Supported databases are:

- **DB2**
- **ORACLE** This value applies to Oracle and Amazon RDS for Oracle
- **MSSQL** This value applies to MSSQL and MSSQL cloud-based databases.
- **POSTGRESQL**

This parameter is required and has no default value.

--dbname *db_name*

The name of the IBM® Workload Scheduler database. This parameter is optional. The default value is **TWS**.

--dbuser *db_user*

The user that has been granted access to the IBM® Workload Scheduler tables on the database server. This parameter is optional. The default value is **db2tws**.

--dbpassword *db_password*

The password for the user that has been granted access to the IBM® Workload Scheduler or Dynamic Workload Console tables on the database server. This parameter is required. The default value is **password**. Special characters are not supported. You can optionally encrypt the password. For more information, see Encrypting passwords (optional) *(on page 39)*.

--dbport *db_port*

The port of the database server. This parameter is optional. The default value is **50000**.

--dbhostname *db_hostname*

The host name or IP address of database server. This parameter is required.

--dbdriverpath *db_driver_path*

The path where the database drivers are stored. This parameter is optional. By default, the configuration script references the JDBC drivers supplied with the product images. If your database server is not compatible with the supplied drivers, then contact your database administrator for the correct version to use with your database server and specify the driver path using this parameter. Ensure you provide the same path in the configureDb, serverinst, and dwcinst commands.

--dbsslconnection *true* | *false*

Enables or disables the SSL connection to the database. The default value is **false**. This parameter applies only to DB2.

SSL configuration options

--sslkeysfolder *keystore_truststore_folder*

The name and path of the folder containing certificates in PEM format. The installation program automatically processes the keystore and truststore files using the password you specify with the **--sslpassword** parameter. The folder must contain the following files:

- **ca.crt**

  The Certificate Authority (CA) public certificate. Note that if certificates being installed are part of a chain consisting of 3 or more certificates (one Root CA, followed by one or more Intermediate CAs, followed by the end user certificate), then this file must contain the Root CA certificate only. Any Intermediate CA certificates must be stored in the `additionalCAs` subfolder, which therefore becomes a mandatory subfolder. Each Intermediate CA must be stored in the `additionalCAs` subfolder in its own file.

  Note: From V10.2.3, if certificates being installed are part of a chain, the ca.crt can contain also the intermediate CAs. In this case, it must begin with one or more intermediate CA certificates and end with the Root ca.

- **tls.key**

  The private key of the end user certificate for the instance to be installed.

- **tls.crt**

  The public part of the previous key, that is the end user certificate.

For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).

You can optionally create a subfolder to contain one or more `*.crt` files to be added to the server truststore as trusted CA, whose name must be `additionalCAs`. This can be used for example to add to the list of trusted CAs the certificate of the LDAP server or DB2 server. Additionally, you can store here any intermediate CA certificate to be added to the truststore. The subfolder must be named **additionalCAs**. Note that if the end user certificate being installed in the instance is part of a chain consisting of 3 or more certificates (one Root CA, followed by one or more Intermediate CAs, followed by the end user certificate), then the Intermediate CAs certificates must be stored in the `additionalCAs` subfolder, which therefore becomes a mandatory subfolder. Each Intermediate CA must be stored in the `additionalCAs` subfolder in its own file.

For further information about how to generate custom certificates, see Managing certificates using Certman *(on page)*.

--sslpassword *ssl_password*

The password for the custom certificates and the path to the folder containing certificates in PEM format with the **sslkeysfolder** parameter.

For more information, see sslkeysfolder *(on page 316)*.

You can optionally encrypt the password using the secure script. For more information, see Optional password encryption - secure script *(on page 300)*.

--enablefips false

Specify whether you want to enable FIPS. In the current product version, you can only specify `false` because FIPS is not supported. In a fresh installation, the default is `false`. In upgrade, there is no default value, so you have to set it explicitly and be aware that FIPS is being disabled when you upgrade. This parameter is optional. If you are upgrading from an environment where FIPS is supported, see Q: My environment is FIPS compliant. What happens if I upgrade to version 10.2.3? *(on page 270)*.

User information

--wauser *user_name*

The user for which you are installing IBM Workload Scheduler. This parameter is optional. The default value is the user performing the installation, unless you use a **user other than root**.

On UNIX operating systems, you can choose to install as the **root** user or as a **user other than root**. The following considerations apply:

- If the installer is the **root user**, the **wauser** parameter can be omitted if the *username* value is meant to be root, or can be set to a *username* value other than root.
- If the installer is **different from the root user**, consider the following points:
  - The wauser parameter can be omitted, but wauser is automatically set to the login name of the installer. If the installer specifies a wauser with a different *username* value, an error message is returned.
  - As a consequence, you can log in to the master domain manager uniquely with the user name of the installer.
  - The user must be enabled to login to the machine where the master domain manager is going to be installed.
  - Event Management triggers on files work only if the selected files are accessible to the user that was used for the installation.
  - Future upgrades, modifications, and removal of the master domain manager can be made exclusively with the same login used for installation.
  - When running **conman** and **composer** commands, it is mandatory to set the environment first, by using the tws_env script as described in Setting the environment variables *(on page 138)*.

--wapassword *wauser_password*

The password for the user for which you are installing IBM Workload Scheduler.

On Windows operating systems

Supported characters for the password are alphanumeric, dash (-), underscore (_) characters, and ()|?*~+.@!^

On UNIX operating systems

Supported characters for the password are any alphanumeric, dash (-), underscore (_) characters, and ()|?=*~+.

This parameter is required if you specify the **wauser** parameter. You can optionally encrypt the password using the secure script. For more information, see Optional password encryption - secure script *(on page 300)*.

Configuration information for the application server

The values for these parameters must match the values defined when installing WebSphere Application Server Liberty Base. For more information, see Installing WebSphere Application Server Liberty Base *(on page 37)*.

--wlpdir | w *wlp_directory*

WebSphere Application Server Liberty Base profile installation directory. This parameter is required.

--httpsport *https_port*

The HTTPS port. This parameter is optional. The default value is **31116**.

--startserver true | false

Specifies whether the WebSphere Application Server Liberty Base server must be started after installation. This parameter is optional. The default value is **true**.

Configuration information for dynamic scheduling

--displayname *agent_name*

The name to be assigned to the agent. The name cannot start with a number. If the host name starts with a number, this parameter is required, otherwise it is optional. The default value is the host name of the workstation followed by **_1**.

--jmport *port_number*

The JobManager port number on which the dynamic domain manager is contacted by the dynamic agent. This parameter is optional. The default value is **31114**. The valid range is from 1 to 65535.

Configuration information for the master domain manager

--componenttype *MDM* | *DDM*

The workstation type being installed. Supported workstation types are:

MDM

master domain manager

DDM

dynamic domain manager

To install a backup domain manager, run the serverinst command on the workstation where you plan to install the backup domain manager. The serverinst command connects to the database you specify, discovers that a master domain manager is already installed, and proceeds to install a backup domain manager. The same procedure applies when installing a backup dynamic domain manager.

Configuration options when --componenttype is MDM

--company *company_name*

The name of the company. The company name cannot contain blank characters. The name is shown in program headers and reports. This parameter is optional. The default name is **COMPANY**.

--hostname *host_name*

The fully qualified host name or IP address on which the installation is performed. The default value is calculated at installation time.

--thiscpu *workstation*

The name of the IBM Workload Scheduler workstation for this installation. The name cannot exceed 16 characters, cannot start with a number, cannot contain spaces. If the host name starts with a number, this parameter is required, otherwise it is optional. This name is registered in the `localopts` file. The default name is the host name of the workstation.

--eifport *eif_port*

Specifies the Job Manager Event Integration Facility (EIF) port number. The default value is **31131**. The valid range is 1 to 65535.

--brwksname *broker_workstation_name*

The broker workstation name. This parameter is optional. The default value is the workstation host name followed by **_DWB**. It cannot start with a number.

--brnetmanport *port_number*

The TCP/IP port number used by the `netman` process to listen for communication from the dynamic domain manager. This parameter is optional. The default value is **41114**. The valid range is from 1 to 65535. This port number is registered in the `localopts` file. For each installation you must specify a different number. For more information about the `localopts` file, see Setting local options *(on page)*

--netmanport *netman_port_number*

The TCP/IP port number used by the `netman` process to listen for communication from the master domain manager. This parameter is optional. The default value is **31111**. The valid range is from 1 to 65535. You can also set this parameter to `disabled`. In this case, you must provide a value for the **netmansslport** parameter, which enables SSL communication. This port number is registered in the `localopts` file, in the **nm port** attribute. For each installation you must specify a different number.

--netmansslport ***SSL_port_number***

The TCP/IP port number used by the netman process to listen for communication from the master in SSL mode. The default value is 31113. The valid range is from 1 to 65535. You can also set the **netmansslport** parameter to `disabled` to use non-encrypted communication. If you set the **netmansslport** parameter to `disabled`, you must provide a value for the **netmanport** parameter. This port number is registered in the `localopts` file, in the **nm ssl full port** attribute. For each installation you must specify a different number.

Configuration options when --componenttype is DDM

--domain ***domain_name***

Windows™ systems only. The domain name of the IBM Workload Scheduler user. This parameter is optional. The default value is **MASTERDM** when you install a master domain manager, and **DYNAMICDM** when you install a dynamic domain manager.

--master ***mdm_domain_name***

The master domain manager name. It cannot start with a number. This parameter is required for the dynamic domain manager only. Do not specify when installing the master domain manager.

--mdmhttpsport ***mdm_https_port***

The port of the master domain manager host used by the broker to contact master domain manager. This parameter is required. This parameter applies to the dynamic domain manager only. Do not specify when installing the master domain manager.

--mdmbrokerhostname ***mdm_hostname***

The fully qualified host name or IP address of the master domain manager contacted by the dynamic domain manager. This parameter is required for the dynamic domain manager only. Do not specify when installing the master domain manager.

--eifport ***eif_port***

Specifies the Job Manager Event Integration Facility (EIF) port number. The default value is **31131**. The valid range is 1 to 65535.

--brwksname ***broker_workstation_name***

The broker workstation name. This parameter is optional. The default value is the workstation host name followed by **_DWB**. It cannot start with a number.

--brnetmanport ***port_number***

The TCP/IP port number used by the `netman` process to listen for communication from the dynamic domain manager. This parameter is optional. The default value is **41114**. The valid range is from 1 to 65535. This port number is registered in the `localopts` file. For each installation you must specify a different number. For more information about the `localopts` file, see Setting local options *(on page)*

--netmanport ***netman_port_number***

The TCP/IP port number used by the `netman` process to listen for communication from the master domain manager. This parameter is optional. The default value is **31111**. The valid range is from 1 to 65535. You can also set this parameter to `disabled`. In this case, you must provide a value for the **netmansslport** parameter, which enables SSL communication. This port number is registered in the `localopts` file, in the **nm port** attribute. For each installation you must specify a different number.

--netmansslport ***SSL_port_number***

The TCP/IP port number used by the netman process to listen for communication from the master in SSL mode. The default value is 31113. The valid range is from 1 to 65535. You can also set the **netmansslport** parameter to `disabled` to use non-encrypted communication. If you set the **netmansslport** parameter to `disabled`, you must provide a value for the **netmanport** parameter. This port number is registered in the `localopts` file, in the **nm ssl full port** attribute. For each installation you must specify a different number.

--isforzos *yes*|*no*

Set to **yes** if you want to connect the dynamic domain manager to only the Z controller. Set to **no** if you want to connect the dynamic domain manager to a master domain manager or, to both a master domain manager and a Z controller. This parameter is optional. The default value is **no**.

IBM® Workload Scheduler encryption options

--useencryption *true* | *false*

Specifies whether IBM® Workload Scheduler files must be encrypted at runtime. If you specify *true*, or do not set this parameter, files such as the Symphony file and the message queues are encrypted using AES-256 or AES-128 cryptography. By default, a fresh installation is automatically encrypted and the keystore password is *default*. To change the keystore password, use the **encryptionpassword** parameter. This parameter is optional.

--encryptionpassword *default*

The password for the keystore storing the AES-256 or AES-128 keys used to encrypt the files at runtime. This parameter is optional. The default value is *default*. You can optionally encrypt the password using the secure script. For more information, see Optional password encryption - secure script *(on page 300)*.

Comments

Note: The values for the following parameters must match the values you provided when creating and populating the database:

- **--rdbmstype**
- **--dbhostname**
- **--dbport**
- **--dbname**
- **--dbuser**
- **--dbpassword**

Dynamic Workload Console installation - dwcinst script

This script installs the Dynamic Workload Console

This section lists and describes the parameters that are used when running a **dwcinst** script to install the Dynamic Workload Console. For a typical installation scenario, see Installing the Dynamic Workload Console servers *(on page 77)*. If you are installing in a z/OS environment, see Installing the Dynamic Workload Console *(on page)*.

Certificates are now required when installing or upgrading IBM® Workload Scheduler. You can no longer install nor upgrade IBM® Workload Scheduler without securing your environment with certificates. The required certificates are:

- ca.crt
- tls.key
- tls.crt

For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).

You can specify values in the properties file, type them in the command line, or use both methods. If a parameter is specified both in the properties file and in the command line, the command line value takes precedence.

Note: To avoid installation failure, ensure that the **inst_dir** parameter is different from the directory of the installation image.

The log files generated from this command are located in the following path:

On Windows operating systems

DWC_home\logs

On UNIX operating systems

DWC_DATA_dir/installation/logs

On z/OS operating system

DWC_DATA_dir/installation/logs

Syntax for Windows operating systems

Show command usage

```
dwcinst -? | --usage | --help
```

Retrieve the command parameters and values from a properties file

```
dwcinst --file | -f  [properties_file]
```

General information

```
 dwcinst
 --acceptlicense yes|no
[--lang lang_id]
[--inst_dir install_dir]
[--skipcheckprereq true|false]
[--componenttype DWC | FED]
```

Configuration information for the data source

```
 --rdbmstype|-r DB2 | DB2Z | ORACLE | MSSQL | POSTGRESQL
 [--dbname db_name]
 [--dbuser db_user]
 [--dbpassword db_password]
 [--dbport db_port]
 [--dbhostname db_hostname]
 [--dbdriverpath db_driver_path]
 [--dbsslconnection true | false]
```

DB2 for z/OS-only configuration options

```
[--zlocationname  zOS_location_containing_db]
```

SSL configuration options

```
--sslkeysfolder  keystore_truststore_folder
--sslpassword ssl_password
```

User information

```
 --user | -u dwc_user
 --password | -p dwc_password
```

Configuration information for the application server

```
 --wlpdir|-w wlp_directory
```

Security configuration

```
    [--httpsport https_port]
    [--bootstrapport bootstrap_port]
    [--bootsecpport bootstrap_sec_port]
```

Syntax for UNIX operating systems

Show command usage

```
dwcinst -? | --usage | --help
```

Retrieve the command parameters and values from a properties file

```
dwcinst --file | -f  [properties_file]
```

General information

```
   dwcinst
   --acceptlicense yes|no
  [--lang lang_id]
  [--inst_dir install_dir]
  [--data_dir dwc_datadir]
  [--skipcheckprereq true|false]
  [--componenttype DWC | FED]
```

Configuration information for the data source

```
   --rdbmstype|-r DB2 | DB2Z | ORACLE | MSSQL | POSTGRESQL
   [--dbname db_name
   [--dbuser db_user]
   [--dbpassword db_password]
   [--dbport db_port]
   [--dbhostname db_hostname]
   [--dbdriverpath db_driver_path]
   [--dbsslconnection true | false]
```

DB2 for z/OS-only configuration options

```
[--zlocationname  zOS_location_containing_db]
```

SSL configuration options

```
  --sslkeysfolder  keystore_truststore_folder
  --sslpassword ssl_password
```

User information

```
    --user | -u dwc_user
    --password | -p dwc_password
```

Configuration information for the application server

```
   --wlpdir|-w wlp_directory
```

Security configuration

```
   [--httpsport https_port]
   [--bootstrapport bootstrap_port]
   [--bootsecpport bootstrap_sec_port]
```

Syntax for z/OS operating systems

Show command usage

```
dwcinst -? | --usage | --help
```

Retrieve the command parameters and values from a properties file

```
dwcinst --file | -f  [properties_file]
```

General information

```
 dwcinst
 --acceptlicense yes|no
[--lang lang_id]
[--inst_dir install_dir]
[--data_dir dwc_datadir]
[--componenttype DWC | FED]
```

Configuration information for the data source

```
 --rdbmstype|-r DB2 | DB2Z | ORACLE | MSSQL | POSTGRESQL
 [--dbname db_name]
 [--dbuser db_user]
 [--dbpassword db_password]
 [--dbport db_port]
 [--dbhostname db_hostname]
 [--dbdriverpath db_driver_path]
```

DB2 for z/OS-only configuration options

```
 [--zlocationname  zOS_location_containing_db]
```

SSL configuration options

```
--sslkeysfolder  keystore_truststore_folder
--sslpassword ssl_password
```

User information

```
  --user | -u dwc_user
  --password | -p dwc_password
```

Configuration information for the application server

```
 --wlpdir|-w wlp_directory
```

Security configuration

```
 [--httpsport https_port]
 [--bootstrapport bootstrap_port]
 [--bootsecpport bootstrap_sec_port]
```

Parameters

-? | -usage | -help

Displays the command usage and exits.

--propfile | -f [*properties_file*]

Optionally specify a properties file containing custom values for `dwcinst` parameters. The default file is located in the root directory of the installation image.

Specifying a properties file is suggested if you have a high number of parameters which require custom values. You can also reuse the file with minimal modification for several installations. If you create a custom properties file, specify its name and path with the **-f** parameter.

General information

--acceptlicense *yes*|*no*

Specify whether to accept the License Agreement.

--lang *lang_id*

The language in which the messages returned by the command are displayed. If not specified, the system LANG is used. If the related catalog is missing, the default C language catalog is used. If neither **--lang** nor LANG are used, the default codepage is set to SBCS. For a list of valid values for these variables, see the following table:

Table 26. Valid values for -lang and LANG parameter

Language	Value
Brazilian Portuguese	pt_BR
Chinese (traditional and simplified)	zh_CN, zh_TW
English	en
French	fr
German	de
Italian	it
Japanese	ja
Korean	ko
Russian	ru
Spanish	es

Note: This is the language in which the installation log is recorded and not the language of the installed component instance. The command installs all languages as default.

--inst_dir

Specify the directory where the Dynamic Workload Console is to be installed. This parameter is optional. The default values varies based on the operating system, as follows:

On Windows operating systems

```
%ProgramFiles%\wa\DWC
```

On UNIX operating systems

```
/opt/wa/DWC
```

On z/OS operating system

```
/opt/wa/DWC
```

After installing, you can find this value in the `twainstance<instance_number>.TWA.properties` file, by checking the

DWC_basePath parameter. For more information, see Finding out what has been installed in which IBM Workload Automation instances *(on page 25)*.

--data_dir ***dwc_datadir***

Specify the path to a directory where you want to store the logs and configuration files produced by Dynamic Workload Console. This parameter is optional. If you do not specify this parameter, all data files generated by the Dynamic Workload Console are stored in *DWC_home*`/DWC_DATA`. This path is called, in the publications, *DWC_DATA_dir*.

--skipcheckprereq true | false

If you set this parameter to `false`, Dynamic Workload Console does not scan system prerequisites before starting the installation. This parameter is optional. The default value is `true`. For more information about the prerequisite check, see Scanning system prerequisites for IBM Workload Scheduler *(on page 33)*.

Configuration information for the data source

--rdbmstype|-r ***rdbms_type***

The database type. Supported databases are:

- **DB2**
- **ORACLE**
- **MSSQL** This value applies to MSSQL and supported MSSQL cloud-based databases.
- **POSTGRESQL**
-

This parameter is required and has no default value.

--dbname ***db_name***

The name of the Dynamic Workload Console database. This parameter is optional. The default value is **DWC**.

--dbuser ***db_user***

The user that has been granted access to the Dynamic Workload Console tables on the database server. This parameter is required.

--dbpassword ***db_password***

The password for the user that has been granted access to the Dynamic Workload Console tables on the database server. This parameter is required. Special characters are not supported. You can optionally encrypt the password. For more information, see Encrypting passwords (optional) *(on page 39)*.

--dbport ***db_port***

The port of the database server. This parameter is required.

--dbhostname ***db_hostname***

The host name or IP address of database server. This parameter is required.

--dbdriverpath ***db_driver_path***

The path where the database drivers are stored. This parameter is optional. By default, the configuration script references the JDBC drivers supplied with the product images. If your database server is not compatible with the supplied drivers, then contact your database administrator for the correct version to use with your database server and specify the driver path using this parameter. Ensure you provide the same path in the configureDb, serverinst, and dwcinst commands.

--dbsslconnection true | false

Enables or disables the SSL connection to the database. This value must always be **false** when `--rdbmstype` is DB2Z.

The default value is **false**.

SSL configuration options

--sslkeysfolder ***keystore_truststore_folder***

The name and path of the folder containing certificates in PEM format. The installation program automatically processes the keystore and truststore files using the password you specify with the **--sslpassword** parameter. The folder must contain the following files:

- **ca.crt**

  The Certificate Authority (CA) public certificate. Note that if certificates being installed are part of a chain consisting of 3 or more certificates (one Root CA, followed by one or more Intermediate CAs, followed by the end user certificate), then this file must contain the Root CA certificate only. Any Intermediate CA certificates must be stored in the `additionalCAs` subfolder, which therefore becomes a mandatory subfolder. Each Intermediate CA must be stored in the `additionalCAs` subfolder in its own file.

  Note: From V10.2.3, if certificates being installed are part of a chain, the ca.crt can contain also the intermediate CAs. In this case, it must begin with one or more intermediate CA certificates and end with the Root ca.

- **tls.key**

  The private key of the end user certificate for the instance to be installed.

- **tls.crt**

  The public part of the previous key, that is the end user certificate.

For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).

You can optionally create a subfolder to contain one or more `*.crt` files to be added to the server truststore as trusted CA, whose name must be `additionalCAs`. This can be used for example to add to the list of trusted CAs the certificate of the LDAP server or DB2 server. Additionally, you can store here any intermediate CA certificate to be added to the truststore. The subfolder must be named **additionalCAs**. Note that if the end user certificate being installed in the instance is part of a chain consisting of 3 or more certificates (one Root CA, followed by one or more Intermediate CAs, followed by the end user certificate), then the Intermediate CAs certificates must be stored in the `additionalCAs` subfolder, which therefore becomes a mandatory subfolder. Each Intermediate CA must be stored in the `additionalCAs` subfolder in its own file.

For further information about how to generate custom certificates, see Managing certificates using Certman *(on page)*.

--sslpassword *ssl_password*

The password for the custom certificates and the path to the folder containing certificates in PEM format with the **sslkeysfolder** parameter.

For more information, see sslkeysfolder *(on page 325)*.

You can optionally encrypt the password using the secure script. For more information, see Optional password encryption - secure script *(on page 300)*.

--enablefips false

Specify whether you want to enable FIPS. In the current product version, you can only specify `false` because FIPS is not supported. In a fresh installation, the default is `false`. In upgrade, there is no default value, so you have to set it explicitly and be aware that FIPS is being disabled when you upgrade. This parameter is optional. If you are upgrading from an environment where FIPS is supported, see Q: My environment is FIPS compliant. What happens if I upgrade to version 10.2.3? *(on page 270)*.

DB2 for z/OS-only configuration syntax

--zlocationname *zos_location_containing_db*

The name of an already existing location in the z/OS environment that will contain the new database. The default value is LOC1.

User information

--user

Specify the administrator of the Dynamic Workload Console. You can use this account to log in to the Dynamic Workload Console and manage your environment. This parameter is optional. The default value is `dwcadmin`.

--password

Specify the password for the Dynamic Workload Console user. This parameter is required. You can optionally encrypt the password. For more information, see Encrypting passwords (optional) *(on page 39)*.

On Windows operating systems

Supported characters for the password are alphanumeric, dash (-), underscore (_) characters, and ()|?*~+.@!^

On UNIX operating systems

Supported characters for the password are any alphanumeric, dash (-), underscore (_) characters, and ()|?=*~+.

Configuration information for the application server

--wlpdir

Specify the path where WebSphere Application Server Liberty Base is installed. This parameter is required.

On z/OS operating system

Specify the path where WebSphere Application Server for z/OS Liberty is installed. This parameter is required.

Security configuration

--httpsport

Specify the HTTPS port, to be used in the Dynamic Workload Console URL. This parameter is optional. The default value is `9443`.

--bootstrapport

Specify the bootstrap port. This parameter is optional. The default value is `12809`.

--bootsecport

Specify the bootstrap security port, to be used for connecting to the Z connector. This parameter is optional. The default value is `19402`.

Agent installation parameters - twsinst script

Agent installation parameters that can be passed to the twsinst script.

This section lists and describes the parameters that are used when running a twsinst script to install dynamic agents, fault-tolerant agents.

Certificates are now required when installing or upgrading IBM® Workload Scheduler. You can no longer install nor upgrade IBM® Workload Scheduler without securing your environment with certificates. The required certificates are:

- ca.crt
- tls.key
- tls.crt

For UNIX systems, ensure that all the files have the ownership of the user who installed the master domain manager and the correct permissions (644).

To find some sample agent installation scenarios, see Example installation commands *(on page 92)* and Dynamic agent gateway installation examples *(on page 94)*.

-acceptlicense *yes*|*no*

Specify whether to accept the License Agreement.

-addjruntime *true|false*

Adds the Java™ run time to run job types with advanced options, both those types that are supplied with the product and the additional types that are implemented through the custom plug-ins. Valid values are **true** and **false**. The default for a fresh installation is **true**. Set this parameter to `true` if you use the **sslkeysfolder** and **sslpassword** parameters to define custom certificates in PEM format.

If you decided not to install Java™ run time at installation time, you can still add this feature later as it is described in Adding a feature *(on page 152)*.

-agent *dynamic|fta|both*

The type of agent that you want to install. Valid values are:

dynamic

To install the IBM Workload Scheduler dynamic agent. Use this value with the **-tdwbhostname** *host_name* and the **-tdwbport** *tdwbport_number* parameters.

On Windows operating systems, you can install dynamic agents using the Local System Account. To install with the Local System Account, omit the **uname** and **password** parameters.

fta

To install the IBM Workload Scheduler fault-tolerant agent.

both

To install the dynamic agent that is used with the **-tdwbhostname** *host_name* and the **-tdwbport** *tdwbport_number* parameters, and a fault-tolerant agent.

The default is ***dynamic***.

-agentid *agent_id*

The unique identifier of the agent that you want to install. The parameter is optional. If not specified, the installation process assigns to the agent a string of alphanumeric characters, as in the following example:

```
893164748CCA4FC6820F12685AECBB07
```

It might be useful to specify an *agent_id* when you want to reinstall an agent after it was uninstalled, and you want to use the same *agent_id*. This prevents that two different *agent_id* values are registered on the server for the same agent installation.

Note: When you manually specify the agent_id value, ensure that the length is 32 characters, otherwise an error occurs.

If you set the **jwt** parameter to `true`, the **agentId** parameter is ignored if provided, because the agent ID is retrieved from the master domain manager together with the JWT. See -jwt true | false *(on page 330)*.

-apikey

Use this parameter to specify the API key to be used for authenticating with the master domain manager. This authentication enables downloading the certificates or JWT to be used for communication between dynamic agent and dynamic domain manager. This parameter is mutually exclusive with the **wauser** and **wapassword** parameters. A random password in base64 encoding is automatically created for generating stash files. The password stored in the `tls.sth` file. If needed, you can decrypt this password using any base64 decoder.

Obtain the string to be provided with this parameter from the Dynamic Workload Console before running the command. For more information, see Authenticating the command line client using API Keys *(on page)*.

-company *company_name*

The name of the company. The company name cannot contain blank characters. The name is shown in program headers and reports. If not specified, the default name is COMPANY.

-create_link

UNIX™ systems only. Create the **symlink** between `/usr/bin/at` and *`install_dir`*`/TWS/bin/at`. For more information, see Table 2: Symbolic link options *(on page 22)*.

-data_dir *path*

This argument applies to UNIX operating systems only. Specify a path for product data, such as log and configuration files, if you want to install the product binaries separated from the product data. This argument is optional. The default value is *`INSTALL_DIR`*`/TWSDATA`.

-displayname *display_name*

The name to assign to the agent. The name cannot start with a number. The default is based on the host name of this computer.

If the host name starts with a number, the **-displayname** parameter must be specified.

-domain *user_domain*

Windows™ systems only. The domain name of the IBM Workload Scheduler user. The default is the name of the workstation on which you are installing the product. Ensure you use `USERDOMAIN` instead of `USERDNSDOMAIN`.

-enablefips *true*|*false*

Specify whether you want to enable FIPS. In the current product version, you can only specify `false` because FIPS is not supported. In a fresh installation, the default is `false`. In upgrade, there is no default value, so you have to set it explicitly and be aware that FIPS is being disabled when you upgrade. This parameter is optional. If you are upgrading from an environment where FIPS is supported, see Q: My environment is FIPS compliant. What happens if I upgrade to version 10.2.3? *(on page 270)*.

-encryptionpassword *default*

The password for the keystore storing the AES-256 or AES-128 keys used to encrypt the files at runtime. This parameter is optional. The default value is *default*. You can optionally encrypt the password using the secure script. For more information, see Optional password encryption - secure script *(on page 300)*.

-gateway *local*|*remote*|*none*

Specifies whether to configure a gateway to communicate with the dynamic workload broker or not, and how it is configured. Specify *`local`* if the gateway is local to the dynamic agent workstation. Specify *`remote`* if the dynamic agent communicates through a gateway that is installed on a different dynamic agent workstation from the dynamic agent being installed. The default value is *`none`*, which means no gateway is configured. For information about installing with a local and remote gateway, see Example installation commands *(on page 92)*.

-gweifport *gateway_eif_port*

Specifies the Job Manager Event Integration Facility (EIF) port number. The default value is **31132**. The valid range is 1 to 65535.

-gwid *gateway_id*

The unique identifier for the gateway. This parameter is required when you specify **-gateway** *`local`*. The default gateway identifier that is assigned is **GW1**. The gateway identifier must start with either an alphabetic character or an underscore character (_), and it can contain only the following types of characters: alphabetic, numeric, underscores (_), hyphens (-), and periods (.).

Gateways can also work in parallel to mutually take over in routing communications to the agents connected to them. To enable gateways to work in parallel, all gateways must have the same *gateway_id* assigned. This information is stored in the `JobManagerGW.ini` file, by setting the JobManagerGWURIs property.

-hostname *host_name*

The fully qualified hostname or IP address on which the agent is contacted by the dynamic workload broker. The default is the hostname of this computer. If the hostname is a localhost, the hostname parameter must be specified.

-inst_dir *installation_dir*

The directory of the IBM Workload Scheduler installation.

On Windows™ operating systems:

If you specify a path that contains blanks, enclose it in double quotation marks. Specify an absolute path. If you do not manually specify a path, the path is set to `%ProgramFiles%\IBM\TWA_`*`TWS_USER`*, where *TWS_USER* is the user for which you are installing the IBM Workload Scheduler that you specify in the **-uname** parameter. If you use the Local System Account and therefore do not specify the **-uname** parameter, the path is set to `%ProgramFiles%\IBM\TWA_WaLocalSystemAccount`.

On UNIX™ and Linux™ operating systems:

If you specify a path that contains blanks, enclose it in double quotation marks. Specify an absolute path. If you do not manually specify a path, the path is set to:

- `/opt/IBM/TWA_`*`TWS_USER`*, if you logged in as the **root** user to install the agent. *TWS_USER* is the user that you specify in the `-uname` option and for which you are installing the agent (can omit if *TWS_USER* is **root**). The IBM Workload Scheduler user that you specify in the `-uname username` parameter must have read and run privileges for the `installation_dir` installation path; otherwise the installation fails.
- *`home_dir`*`/TWA`, if you logged in with a login **other than root**. Ensure that the directory permission is set to **755** for *home_dir*, the home directory for your login, and that you are the *home_dir* owner.

-jmport ***port_number***

The JobManager port number used by the dynamic workload broker to connect to the dynamic agent. The default value is **31114**. The valid range is from 1 to 65535.

-jmportssl ***true|false***

The JobManager port used by the dynamic workload broker to connect to the IBM Workload Scheduler dynamic agent. The port value is the value of the ssl_port parameter in the `ita.ini` file if **-jmportssl** is set to `true`. If set to `false`, it corresponds to the value of the **tcp_port** parameter in the `ita.ini` file. The `ita.ini` file is located in `ITA\cpa\ita` on Windows™ systems and `ITA/cpa/ita` on UNIX™, Linux™, and IBM i systems.

Set the value to "true" if **- gateway** is set to `local`.

For communication using SSL or HTTPS

Set **jmportssl = *true***. To communicate with the dynamic workload broker, it is recommended that you set the value to *true*. In this case, the port specified in **jmport** communicates in HTTPS.

For communication without using SSL or through HTTP

Set **jmportssl = *false***. In this case the port specified in **jmport** communicates in HTTP.

-jwt ***true*** **|** ***false***

Specify `true` to use the JSON Web Token (JWT) to authenticate with the master domain manager. Specify `false` to authenticate with the master domain manager using certificates. The default value is `true`. This parameter is mutually exclusive with the **sslkeysfolder** and **sslpassword** parameters which are used to generate custom certificates.

If you set this parameter to `true`, the **agentId** parameter is ignored if provided, because the agent ID is retrieved from the master domain manager together with the JWT. See -agentid agent_id *(on page 328)*. Also, if you set this parameter to `true`, the following parameters are required for downloading the JWT:

- **wauser** or **apikey**. See -wauser wauser_name *(on page 334)* or -apikey *(on page 328)*.
- **wapassword** or **apikey**. See -wapassword wauser_password *(on page 334)* or -apikey *(on page 328)*.
- **tdwbhostname**. See -tdwbhostname host_name *(on page 333)*.
- **tdwbport**. See -tdwbport tdwbport_number *(on page 333)*.

For examples of installations with JWT, see Example installation commands *(on page 92)*.

-lang ***lang_id***

The language in which the twsinst messages are displayed. If not specified, the system LANG is used. If the related catalog is missing, the default C language catalog is used. If neither **-lang** nor LANG are used, the default codepage is set to SBCS. For a list of valid values for these variables, see the following table:

Table 27. Valid values for -lang and LANG parameter

Language	Value
Brazilian portuguese	pt_BR
Chinese (traditional and simplified)	zh_CN, zh_TW
English	en
French	fr
German	de
Italian	it
Japanese	ja
Korean	ko
Russian	ru
Spanish	es

Note: This is the language in which the installation log is recorded and not the language of the installed engine instance. twsinst installs all languages as default.

-master ***workstation***

The workstation name of the master domain manager. This name cannot exceed 16 characters, cannot contain spaces, and cannot be the same as the workstation name that you entered in the **thiscpu** parameter. If not specified, the default value is **MASTER**.

-new

A fresh installation of the agent. Installs an agent and all supported language packs.

-password ***user_password***

Windows™ systems only. The password of the user for which you are installing IBM Workload Scheduler. The password can include alphanumeric, dash (-), and underscore (_) characters, and the following symbols: ()!? =^*/~ [] $`+;:.@. The **-password** parameter is used for fresh installations only, it is not required for fix packs or upgrades. You can optionally encrypt the password using the secure script. For more information, see Optional password encryption - secure script *(on page 300)*.

On Windows operating systems, you can install dynamic agents using the Local System Account. To install with the Local System Account, omit the **uname** and **password** parameters.

-netmansslport ***SSL_port_number***

The TCP/IP port number used by the netman process to listen for communication from the master in SSL mode. The default value is 31113. The valid range is from 1 to 65535. You can also set the **netmansslport** parameter to `disabled` to use non-encrypted communication. If you set the **netmansslport** parameter to `disabled`, you must provide a value for the **netmanport** parameter. This port number is registered in the `localopts` file, in the **nm ssl full port** attribute. For each installation you must specify a different number.

-port ***port_number***

The TCP/IP port number used by the Netman process to listen for communication from the master. The default value is **31111**. The valid range is from 1 to 65535. This port number is registered in the `localopts` file. For each installation you must specify a different number.You can also set this parameter to `disabled`. In this case, you must provide a value for the **netmansslport** parameter, which enables SSL communication.

-reset_perm

UNIX™ and IBM i systems only. Reset the permission of the libraries in the `/usr/ibm` directory.

-restore

Run this command from the folder to where you copied the eImage (a folder other than the home directory of *TWS_USER*, where *TWS_USER* is the user that installed the IBM Workload Scheduler instance), and not from the installation path, to restore the version in the eImage.

-skip_usercheck

Enable this option if the authentication process within your organization is not standard, thereby disabling the default authentication option.

On Windows™ systems, if you specify this parameter, the program does not create the user you specified in the **-uname** *username* parameter and you must create the user manually before running the script. However, if you use Local System Account, you do not need to specify any user.

On UNIX™ and Linux™ systems if you specify this parameter, the program skips the check of the user in the `/etc/passwd` file or the check you perform using the su command.

-skipcheckprereq

If you specify this parameter, IBM Workload Scheduler does not scan system prerequisites before installing the agent. For more information on the prerequisite check, see Scanning system prerequisites for IBM Workload Scheduler *(on page 33)*.

-sslkeysfolder *path*

The name and path of the folder on the agent containing PEM certificates. The installation program automatically generates the keystore and truststore files using the password you specify with the **--sslpassword** parameter.

The folder must contain the following files and folders:

ca.crt

The Certificate Authority (CA) public certificate.

tls.key

The private key for the instance to be installed.

tls.crt

The public part of the previous key.

tls.sth

The file storing your encoded password in Base64 encoding.

You can optionally create a subfolder to contain one or more `*.crt` files to be added to the server truststore as trusted CA. This can be used for example to add to the list of trusted CAs the certificate of the LDAP server or DB2 server. Additionally, you can store here any intermediate CA certificate to be added to the truststore. The subfolder must be named **additionalCAs**. If you are connecting a master domain manager using custom certificates to a dynamic agent also using custom certificates, the only required file is `ca.crt`.

Before you start the installation, ensure the required files and folders are available on the agent.

The **sslkeysfolder** and **sslpassword** parameters are mutually exclusive with the **wauser**, **wapassword**, and **jwt** parameters, which are used to download and deploy the certificates or JWT already available on the master domain manager.

-sslpassword *password*

Specify the password for the certificates in PEM format automatically generated by the installation program.

If you use this parameter, ensure that the **addjruntime** parameter is set to true, because Java™ run time is required for defining custom certificates.

You can optionally encrypt the password using the secure script. For more information, see Optional password encryption - secure script *(on page 300)*.

-tdwbhostname ***host_name***

The fully qualified host name or IP address of the dynamic workload broker the agent is registering to. It applies when you set the **-agent** parameter to either **dynamic** or **both** and requires the **-tdwbport** parameter. This parameter is required.

If you set the **-gateway** parameter to `remote`, this is the host name of the dynamic agent hosting the gateway and to which the agent you are installing will connect. This information is stored in the `JobManager.ini` file. For information about installing with a local and remote gateway, see Example installation commands *(on page 92)*.

If you set the **jwt** parameter to `true`, ensure you provide a value for this parameter, so that you can download the JWT and agent ID from the specified dynamic domain manager. The dynamic domain manager routes the JWT request to the master domain manager. See also -jwt true | false *(on page 330)*.

-tdwbport ***tdwbport_number***

The HTTPS transport port number of the dynamic workload broker the agent is registering to. It must match the port you specified with **httpsport** parameter when installing the master domain manager. It applies when you set the **-agent** parameter to either **dynamic** or **both** and requires the **-tdwbhostname** parameter. The valid range is from 0 to 65535. If you specify `0` you cannot run workload dynamically. Do not specify `0` if the *-agent* value is `dynamic` or `both`. The default is `0` for an upgrade, which means that this connection is not configured, otherwise, specify `31116` for a fresh installation. This parameter is required.

If you set the **-gateway** parameter to `remote`, this is the HTTP or HTTPS port number of the dynamic agent hosting the gateway and to which the agent you are installing will connect. You have specified this port with the **jmport** parameter when installing the agent hosting the gateway. For information about installing with a local and remote gateway, see Example installation commands *(on page 92)*.

If you are performing a fresh installation, the value to use is `31114`. This information is stored in the `JobManager.ini` file.

If you set the **jwt** parameter to `true`, ensure you provide a value for this parameter, so that you can download the JWT and agent ID from the specified dynamic domain manager. The dynamic domain manager routes the JWT request to the master domain manager. See also -jwt true | false *(on page 330)*.

-thiscpu ***workstation***

The name of the IBM Workload Scheduler workstation of this installation. The name cannot exceed 16 characters, cannot start with a number, cannot contain spaces, and cannot be the same as the workstation name of the master domain manager. This name is registered in the `localopts` file. If not specified, the default value is the host name of the workstation.

If the host name starts with a number, **-thiscpu** parameter must be specified.

-u

Displays command usage information and exits.

-uname ***username***

The name of the user for which the IBM Workload Scheduler agent is being installed. This user owns the IBM Workload Scheduler instance and by default, jobs are run with its name. This user name is not to be confused with the user performing the installation, unless you use a **user other than root**. The user name cannot contain periods (.).

On UNIX™ and Linux™ systems, for a new installation, this user account must be created manually before running the installation and must be enabled to login to the machine where the agent is going to be installed. Create a user with a home directory. IBM Workload Scheduler is installed by default under the home directory of the specified user.

You can choose to install agents as the **root** user, or as a **user other than root**. The following considerations apply:

- If the installer is the **root user**, the **uname** parameter can be omitted if the *username* value is meant to be root, or can be set to a username value other than root.
- If the installer is **different from the root user**, consider the following points:

- The uname parameter can be omitted, but *username* is automatically set to the login name of the installer. If the installer specifies a uname with a different *username* value, an error message is returned.
- As a consequence, the agent can run jobs uniquely with the user name of the installer.
- The user must be enabled to login to the machine where the agent is going to be installed.
- Event Management triggers on files work only if the selected files are accessible to the user that was used for the installation.
- Future upgrades, modifications, and removal of the agent can be made exclusively with the same login used for installation. For dynamic agents, the login name used by the installer is stored in the read-only `InstallationLoginUser` parameter in the `JobManager.ini`configuration file on the agent.
- When running **conman** and **composer** commands, it is mandatory to set the environment first, by using the tws_env script as described in Setting the environment variables *(on page 138)*.

On Windows operating systems, you can install dynamic agents using the Local System Account. To install with the Local System Account, omit the **uname** and **password** parameters.

-useencryption true | false

Specifies whether IBM® Workload Scheduler files must be encrypted at runtime. If you specify *true*, or do not set this parameter, files such as the Symphony file and the message queues are encrypted using AES-256 or AES-128 cryptography. By default, a fresh installation is automatically encrypted and the keystore password is *default*. To change the keystore password, use the **encryptionpassword** parameter. This parameter is optional.

-wauser *wauser_name*

One of the following users, defined on the master domain manager:

- The user for which you have installed the master domain manager the agent is connecting to.
- The user with the DISPLAY permission on the FILE named AGENT_CERTIFICATE. This permission allows the user to download certificates or JWT. For more information about this scenario, see Downloading certificates or JWT using a different user *(on page 342)*.

Always specify the user defined on the master domain manager, also if you are installing a dynamic agent and want it to register to a dynamic domain manager. This is because the dynamic domain manager simply forwards data to and from the master domain manager.

By providing the **wauser** and **wapassword** parameters or the **apikey** parameter, you enable IBM Workload Scheduler to download and install either the certificates in PEM format or the JWT already available on the master domain manager. To download PEM certificates, set the **jwt** parameter to `false`, to download JWT, set the **jwt** parameter to `true`. For more information, see -jwt true | false *(on page 330)*.

This parameter is mutually exclusive with the **apikey** parameter, which provides authentication using an API Key.

For more information, see -apikey *(on page 328)*.

This parameter is also mutually exclusive with the **sslkeysfolder** parameter, which is used to specify a folder on the agent where you store the certificates. For more information, see -sslkeysfolder path *(on page 332)*.

For further information about how to automatically download and deploy certificates in PEM format from the master domain manager to dynamic agents and fault-tolerant agents, see Certificates download to dynamic agents and fault-tolerant agents - AgentCertificateDownloader script *(on page 338)*.

-wapassword *wauser_password*

One of the following passwords, defined on the master domain manager:

- The password of the user for which you have installed the master domain manager the agent is connecting to.
- The password of the user with the DISPLAY permission on the FILE named AGENT_CERTIFICATE. This permission allows the user to download certificates or JWT. For more information about this scenario, see Downloading certificates or JWT using a different user *(on page 342)*.

Always specify the user defined on the master domain manager, also if you are installing a dynamic agent and want it to register to a dynamic domain manager. This is because the dynamic domain manager simply forwards data to and from the master domain manager.

By providing the **wauser** and **wapassword** parameters or the **apikey** parameter, you enable IBM Workload Scheduler to download and install either the certificates in PEM format or the JWT already available on the master domain manager. To download PEM certificates, set the **jwt** parameter to `false`, to download JWT, set the **jwt** parameter to `true`.

See also -jwt true | false *(on page 330)*.

This parameter is mutually exclusive with the **apikey** parameter, which provides authentication using an API Key.

For more information, see -apikey *(on page 328)*.

This parameter is also mutually exclusive with the **sslkeysfolder** parameter, which is used to specify a folder on the agent where you store the certificates. You can optionally encrypt the password using the secure script. For more information, see Optional password encryption - secure script *(on page 300)*.

For further information about how to automatically download and deploy certificates in PEM format from the master domain manager to dynamic agents and fault-tolerant agents, see Certificates download to dynamic agents and fault-tolerant agents - AgentCertificateDownloader script *(on page 338)*.

-work_dir ***working_dir***

The temporary directory used by the program to deploy the installation process files.

On Windows™ operating systems:

If you specify a path that contains blanks, enclose it in double quotation marks. If you do not manually specify a path, the path is set to `%temp%\TWA\tws`*`version_number`*, where `%temp%` is the temporary directory of the operating system.

On UNIX™ and Linux™ operating systems:

The path cannot contain blanks. If you do not manually specify a path, the path is set to `/tmp/TWA/tws`*`version_number`*.

This parameter can also function as a backup directory during product upgrade with path *`WORKING_DIR`*`/backup` if you do not set the **-skipbackup** parameter to **true**.

-v

Displays the command version and exits.

File Proxy installation - fileproxyinst script

This script installs the File Proxy in SSL mode

This section lists and describes the parameters that are used when running the **fileproxyinst** script to install the File Proxy in SSL mode on a workstation different from the master domain manager, where it is already installed by default. This command is supported on the following operating systems:

- Windows with hardware x86-64
- Linux with hardware x86-64
- Linux with hardware IBM z Systems

You can optionally install the File Proxy in high availability configuration by specifying one or more proxy servers or a load balancer. To set up this configuration, use the **Broker.fileproxy.urls** property in the `BrokerWorkstation.properties`. For more information, see BrokerWorkstation.properties file *(on page)*.

Log files produced by this command are located in *data_dir*`/logs/`. By default *data_dir* is *`installation_directory`*`/FILEPROXYDATA`.

Syntax

Windows operating systems:

```
fileproxyinst.exe -acceptlicense yes [-lang language] -inst_dir installation_directory
 [-data_dir data_directory]
[-host hostname] [-sslport ssl_port_number] -sslfolder ssl_folder -sslpassword ssl_pwd
[-java_home java_home]
```

Linux64 and Linux for OS/390 operating systems:

```
./fileproxyinst -acceptlicense yes [-lang language] -inst_dir installation_directory
 [-data_dir data_directory]
[-host hostname] [-sslport ssl_port_number] -sslfolder ssl_folder -sslpassword ssl_pwd
[-java_home java_home]
```

Arguments

-acceptlicense `yes|no`

Required. Specify whether to accept the License Agreement. The default is `no`.

-lang *language*

Optional. The language in which the messages returned by the command are displayed. The default value is `en_us`.

-inst_dir *installation_directory*

Required. The directory of the File Proxy installation. The default value is the directory from which you start the command. Ensure that you have write access to this directory.

-data_dir *data_dir*

Optional. The directory where logs and configuration files are stored. The default value is *`installation_directory`*`/FILEPROXYDATA`. Ensure that you have write access to this directory.

-host *hostname*

Optional. The global, public host name of the workstation where you install the File Proxy or the IP address of the workstation.

-sslport *ssl_port_number*

Optional. The port to be used for secure communication. Supported values are integers between 1 and 65535. The default is `44444`.

-sslfolder *ssl_folder*

Required. The name and path of the folder containing the certificates in PEM format. (For details about how to create the certificates, see Creating a Certificate Authority and generating certificates *(on page)*). It must contain the following files and folder:

- `ca.crt`
- `tls.crt`
- `tls.key`
- `additionalCAs` folder

Note: In the z/OS environment:

- When you create the certificate authority by issuing the command `./openssl x509 -req -in tls.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out tls.crt -days xxx`, ensure that you set the appropriate number of days with the -days parameter. The default is 30.
- The `additionalCAs` folder must contain the public key certificate or public certificate chain of the Z controller SSL keyring.

-sslpassword *ssl_pwd*

Required. The password to access the certificates.

-java_home *java_home*

Optional. The default value is the path to the `.jre` file provided with the product image.

Examples

To specify the installation and data directories and also define a value for the port to be used for SSL communication, for the folder where the certificates are stored, and the password to access them, run the following command:

```
./fileproxyinst -acceptlicense yes -inst_dir /opt/wa/fileproxy -data_dir /opt/wa/mydata -sslport 44445
-sslfolder /opt/wa/my_ssl_folder -sslpassword my_ssl_pwd
```

To specify only the required parameters for the command, defining the folder where the certificates are stored, and the password to access them, run the following command:

```
./fileproxyinst -acceptlicense yes -sslfolder /opt/wa/my_ssl_folder -sslpassword my_ssl_pwd
```

File Proxy start - fileproxystart script

This script starts the File Proxy in SSL mode

This section describes the **fileproxystart** script. This script is used to start the File Proxy in SSL mode on a workstation different from the master domain manager, where it is already installed by default. Use this command to start the File Proxy in case it stops unexpectedly. This command is supported on the following operating systems:

- Windows with hardware x86-64
- Linux with hardware x86-64
- Linux with hardware IBM z Systems

By default, log files produced by this command are located in *`data_dir`*`/logs`.

Syntax

Windows operating systems:

```
fileproxystart.exe
```

Linux:

```
./fileproxystart
```

File Proxy stop - fileproxystop script

This script stops the File Proxy

This section describes the **fileproxystop** script. This script is used to stop the File Proxy. This command is supported on the following operating systems:

- Windows with hardware x86-64
- Linux with hardware x86-64
- Linux with hardware IBM z Systems

Syntax

Windows operating systems:

```
fileproxystop.exe
```

Linux:

```
./fileproxystop
```

File Proxy uninstallation - uninstall script

This script uninstalls the File Proxy

This section describes the **uninstall** script. This script is used to uninstall the File Proxy. This command is supported on the following operating systems:

- Windows with hardware x86-64
- Linux with hardware x86-64
- Linux with hardware IBM z Systems

By default, log files produced by this command are located in *`data_dir`*`/logs`.

Launch the command from the installation directory.

Syntax

Windows operating systems:

```
uninstall.exe -inst_dir installation_directory [-lang language]
```

Linux:

```
./uninstall -inst_dir installation_directory [-lang language]
```

where

-inst_dir *installation_directory*

is the directory where the File Proxy is installed.

-lang *language*

is the language in which the messages returned by the command are displayed.

Comments

The command removes all data related to the File Proxy and leaves the *data_dir* unchanged.

Certificates download to dynamic agents and fault-tolerant agents - AgentCertificateDownloader script

This script downloads and deploys certificates in PEM format from the master domain manager to dynamic agents and fault-tolerant agents or enables authentication through JSON Web Token (JWT).

You can use this script either to download and deploy certificates in PEM format from the master domain manager to the dynamic agents and fault-tolerant agents in your environment or to enable authentication through JSON Web Token (JWT).

When installing the agent with a fresh installation, you only need to provide the credentials to connect to the master domain manager using the wauser and wapassword or the **apikey** parameters. The certificates in PEM format are automatically downloaded and deployed to the agent without further intervention. The same happens if you are using JWT as authentication method.

Note: If you use a load balancer between the dynamic agent and the master domain manager, and the load balancer uses a Certificate Authority (CA), you must convert the .p12 certificates into the PEM format and replace the existing ones.

When certificates are nearing their expiration time, new certificates are automatically downloaded to agents, but error conditions might happen, for example in case the agent is down or disconnected when the automatic certificate update takes place. In this case, you can use this command if you need to change the PEM certificates after their expiration time, or if you have downloaded wrong PEM certificates.

You can also use this command to obtain a new JWT for an agent from which you had previously revoked the JWT. Fore more information about revoking a JWT, see Revoking and reissuing a JSON Web Token *(on page 147)*.

If you authenticate using PEM certificates, the script connects to the master domain manager to retrieve the compressed file containing the certificates, and saves them to the working directory with name `waCertificates.zip`.

Before running the command, ensure the certificates in PEM format are available on the master domain manager in one of the following paths:

On Windows operating systems

```
installation_directory\TWS\ssl\depot
```

On UNIX operating systems

```
TWA_DATA_DIR/ssl/depot
```

The required files are:

ca.crt

The Certificate Authority (CA) public certificate.

tls.key

The private key for the instance to be installed.

tls.crt

The public part of the previous key.

tls.sth

The file storing your encoded password in Base64 encoding.

You can optionally create a subfolder to contain one or more `*.crt` files to be added to the server truststore as trusted CA. This can be used for example to add to the list of trusted CAs the certificate of the LDAP server or DB2 server. Additionally, you can store here any intermediate CA certificate to be added to the truststore. The subfolder must be named **additionalCAs**. If you are connecting a master domain manager using custom certificates to a dynamic agent also using custom certificates, the only required file is `ca.crt`.

You can specify values in the properties file, type them in the command line, or use both methods. If a parameter is specified both in the properties file and in the command line, the command line value takes precedence.

After running the command, stop and restart the agent process using the ShutDownLwa and StartUpLwa commands. For more information about the commands, see ShutDownLwa - Stop the agent *(on page)* and StartUpLwa - Start the agent *(on page)*.

Ensure you start the command from a brand-new shell.

Syntax

Certificate installation syntax on Windows operating systems

Show command usage

```
cscript AgentCertificateDownloader.vbs -? | --usage | --help
```

Retrieve the command parameters and values from a properties file

```
cscript AgentCertificateDownloader.vbs --file | -f  [properties_file]
```

General information

```
cscript AgentCertificateDownloader.vbs
--wauser wauser_name
--wapassword wauser_password
[--jwt true]
[--apikey API_key_string]
[--tdwbhostname host_name]
[--tdwbport tdwbport_number]
--work_dir working_dir
[--gateway local | remote | none]
[--gwid gateway_id]
[--displayname agent_name]
```

Certificate installation syntax on UNIX operating systems

Show command usage

```
./AgentCertificateDownloader.sh --? | --usage | --help
```

Retrieve the command parameters and values from a properties file

```
./AgentCertificateDownloader.sh --file | --f  [properties_file]
```

General information

```
./AgentCertificateDownloader.sh
--wauser wauser_name
--wapassword wauser_password
[--jwt true]
[--apikey API_key_string]
[--tdwbhostname host_name]
[--tdwbport tdwbport_number]
--work_dir working_dir
[--gateway local | remote | none]
[--gwid gateway_id]
[--displayname agent_name]
```

AgentCertificateDownloader parameters

--? | --usage | --help

Displays the command usage and exits.

--propfile | --f [*properties_file*]

Optionally specify a properties file containing custom values for AgentCertificateDownloader parameters. The default file is located in the root directory of the installation image.

Specifying a properties file is suggested if you have a high number of parameters which require custom values. You can also reuse the file with minimal modification for several installations. If you create a custom properties file, specify its name and path with the **-f** parameter.

General information

--wauser *wauser_name*

One of the following users, defined on the master domain manager:

- The user for which you have installed the master domain manager the agent is connecting to.
- The user with the DISPLAY permission on the FILE named AGENT_CERTIFICATE. This permission allows the user to download certificates or JWT. For more information about this scenario, see Downloading certificates or JWT using a different user *(on page 342)*.

By providing the **wauser** and **wapassword** parameters or the **apikey** parameter, you enable IBM Workload Scheduler to download and install either the certificates in PEM format or the JWT already available on the master domain manager. To download PEM certificates, set the **jwt** parameter to `false`, to download JWT, set the **jwt** parameter to `true`.

This parameter is always required, unless you specify the **apikey** parameter, which defines a different authentication method.

--wapassword ***wauser_password***

One of the following passwords, defined on the master domain manager:

- The password of the user for which you have installed the master domain manager the agent is connecting to.
- The password of the user with the DISPLAY permission on the FILE named AGENT_CERTIFICATE. This permission allows the user to download certificates or JWT. For more information about this scenario, see Downloading certificates or JWT using a different user *(on page 342)*.

Always specify the user defined on the master domain manager, also if you are installing a dynamic agent and want it to register to a dynamic domain manager. This is because the dynamic domain manager simply forwards data to and from the master domain manager.

By providing the **wauser** and **wapassword** parameters or the **apikey** parameter, you enable IBM Workload Scheduler to download and install either the certificates in PEM format or the JWT already available on the master domain manager. To download PEM certificates, set the **jwt** parameter to `false`, to download JWT, set the **jwt** parameter to `true`.

This parameter is always required, unless you specify the **apikey** parameter, which defines a different authentication method.

--jwt ***true***

Specify `true` to download the JSON Web Token (JWT) to authenticate with the master domain manager. If you specified `true` at installation time, you can no longer change this setting to `false` and switch to PEM certificates. The only supported operation is switching from PEM certificates to JWT authentication by setting this parameter to `true`.

This parameter is mutually exclusive with the **sslkeysfolder** and **sslpassword** parameters which are used to generate custom certificates.

If you set this parameter to `true`, the following parameters are required for downloading the JWT:

- **wauser** or **apikey**
- **wapassword** or **apikey**
- **tdwbhostname**
- **tdwbport**

-apikey

Use this parameter to specify the API key to be used for authenticating with the master domain manager. This authentication enables downloading the certificates or JWT to be used for communication between dynamic agent and dynamic domain manager. This parameter is mutually exclusive with the **wauser** and **wapassword** parameters. A random password in base64 encoding is automatically created for generating stash files. The password stored in the `tls.sth` file. If needed, you can decrypt this password using any base64 decoder.

Obtain the string to be provided with this parameter from the Dynamic Workload Console before running the command. For more information, see Authenticating the command line client using API Keys *(on page)*.

--tdwbhostname ***host_name***

The fully qualified host name or IP address of the broker server to which the agent is connected. The default value is *localhost*. If you set the **jwt** parameter to `true`, ensure you provide a value for this parameter, to download the JWT and agent ID from the dynamic domain manager. The dynamic domain manager routes the JWT request to the master domain manager.

--tdwbport ***tdwbport_number***

Specify the port of the broker server to which the agent is connected. The default value is 31116. If you set the **jwt** parameter to `true`, ensure you provide a value for this parameter, to download the JWT and agent ID from the dynamic domain manager. The dynamic domain manager routes the JWT request to the master domain manager

--work_dir ***working_dir***

The working directory used to store the `waCertificates.zip` file returned by the command. This compressed file contains the certificates in PEM format retrieved from the master domain manager. This parameter is required and no default value is provided.

--gateway ***local|remote|none***

Specifies whether to configure a gateway to communicate with the dynamic workload broker or not, and how it is configured. Specify *`local`* if the gateway is local to the dynamic agent workstation. Specify *`remote`* if the dynamic agent communicates through a gateway that is installed on a different dynamic agent workstation from the dynamic agent being installed. The default value is *`none`*, which means no gateway is configured. For information about installing with a local and remote gateway, see Example installation commands *(on page 92)*.

--gwid ***gateway_id***

The unique identifier for the gateway. This parameter is required when you specify **-gateway** *`local`*. The default gateway identifier that is assigned is **GW1**. The gateway identifier must start with either an alphabetic character or an underscore character (_), and it can contain only the following types of characters: alphabetic, numeric, underscores (_), hyphens (-), and periods (.).

Gateways can also work in parallel to mutually take over in routing communications to the agents connected to them. To enable gateways to work in parallel, all gateways must have the same *gateway_id* assigned. This information is stored in the `JobManagerGW.ini` file, by setting the JobManagerGWURIs property.

--displayname ***agent_name***

Specify the name assigned the agent.

You can also use the **wapassword** and **wauser** parameters to specify a user different from the user which installed the master domain manager by using an ACL, as described in Downloading certificates or JWT using a different user *(on page 342)*.

For more information about the typical installation procedure, see Typical installation scenario *(on page 36)*.

Downloading certificates or JWT using a different user

Procedure to download and deploy certificates or JWT from the master domain manager to agents using a user different from the user which installed the master domain manager.

To define a user different from the user which installed the master domain manager, perform the following steps:

1. Browse to the `authentication_config.xml` file located in:

   On UNIX operating systems

   TWA_DATA_DIR`/usr/servers/engineServer/configDropins/overrides`

   On Windows operating systems

   TWA_home`\usr\servers\engineServer\configDropins\overrides`
2. Create a backup copy of the file to a different directory and add the new user and password to the file in the `overrides` directory.
3. Create a new role for the user, as follows:

```
composer new srol
```

```
SECURITYROLE DOWNLOAD_CERT_SROLE
FILE DISPLAY
END
```

4. Create a new domain for the user, as follows:

   ```
   composer new sdom
   ```

   ```
   SECURITYDOMAIN DOWNLOAD_DOMAIN
   FILE NAME="AGENT_CERTIFICATE"
   END
   ```

5. Create a new access control list for the user, as follows:

   ```
   composer new acl
   ```

   ```
   ACCESSCONTROLLIST FOR DOWNLOAD_DOMAIN
   other_user DOWNLOAD_CERT_SROLE
   END
   ```

   where *other_user* is the user inserted into `authentication_config.xml`.

   You can now use the *other_user*, which has only the DISPLAY role for file `AGENT_CERTIFICATE`, to install the agent and download certificates or JWT, or to run the AgentCertificateDownload script and download and deploy certificates or JWT.

   You can also perform the same operations from the Dynamic Workload Console, as described in Managing Workload Security *(on page)*.

Index

D

E

F

N

V

W

Z